4451

Linux System Administration

www.4linux.com.br

Conteúdo

Compactadores, Empacotadores e Procedimentos de Backup

2.1 Introdução Teórica

A compressão e empacotamento de arquivos e diretórios é muito importante em qualquer sistema computacional. Ambos os procedimentos são necessários desde o ponto de vista de distribuição de softwares, de economia de banda e de espaço de armazenamento, e de backup do sistema. Veremos neste capítulo o principal programa de empacotamento GNU/Linux e os dois principais compactadores.

A forma mais conhecida de realizar compressão e empacotamento em ambiente Windows é utilizando o programa **"Winzip"**. Um programa que **"zipa"** um arquivo, ou diversos arquivos, na realidade está realizando dois procedimentos distintos: Empacotar e comprimir.

Em ambientes "Unix-like", essas duas tarefas são realizadas de forma logicamente distintas.

O programa **"tar"**, cujo nome deriva de **"tape archiver",** realiza a tarefa de concatenar todos os arquivos e diretórios preservando as informações do "filesystem", isto é, seus meta-dados.

Criado com **propósito de backup** em dispositivos de acesso sequencial (unidades de fita), o "tar" é utilizado hoje em dia como uma ferramenta de empacotamento, podendo ser utilizado em conjunto com compactadores como "gzip" ou "bzip2".

A utilização da ferramenta “tar” é bastante simples. Seguindo o filosofia Unix “faça apenas uma tarefa, mas faça bem feito”, o “tar” é um programa especialista em empacotar vários arquivos. Dessa forma, quando utilizamos os parâmetros “z” ou “j” estamos na realidade fazendo uma chamada externa aos comandos **“gzip”** ou “**bzip2”**, especialistas em compressão de dados.

Outros programas que trabalham de forma análoga ao **“tar”** são o **“dump”** e **“cpio”**. Ambos foram criados com a mesma finalidade, mas são pouco utilizados hoje em dia, pois não são tão versáteis quanto o “tar”.

Este capítulo explica muitas coisas sobre compactação e empacotamento de arquivos, tudo isso é extremamente necessário quando falamos de “backup”. Podemos ter diferentes tipos de “backup”, são eles:

- **Incremental -** O “backup” incremental visa salvar apenas as diferenças em relação ao ultimo “backup” completo, por exemplo: Um “backup” completo acontece no domingo. O incremental salvará os dados de domingo para segunda, de domingo para terça, de domingo para quarta, de domingo para quinta, de domingo para sexta e de domingo para sábado, ou seja, até chegar no próximo “backup” completo.

- **Diferencial -**Diferente do incremental, o diferencial, faz apenas os incrementos, assim gerando um volume menor de dados. Se o “backup” completo foi gerado no domingo, ele salva de domingo para segunda, de segunda para terça, de terça para quarta e assim até o próximo “backup” completo chegar.

- **Completo -** Visa salvar todos os dados, mesmo o que já foram salvos anteriormente, assim fazendo um “backup” completo de todos os objetos em questão.

Conhecendo os tipos de “backup”, vamos conhecer algumas ferramentas que podem nos ajudar.

Crie o diretório de backup para os nosso testes:

```
1 # mkdir /backup
```

2.2 O empacotador cpio

O comando cpio empacota arquivos/diretórios, suas principais opções são:

-i -> extrair backup

-o -> criar backup

-t -> mostrar uma tabela com o conteúdo do backup

-F -> essa opção especifica o arquivo de backup

-d -> cria diretórios se necessário

-v -> mostrar o que está fazendo em detalhes

-u -> sobrescreve arquivos existentes Então, vejamos os principais grupos de opções e como utilizá-las. Para empacotar o conteúdo do diretório "/etc" com o "cpio" devemos fazer o seguinte:

```
1 #  find /etc | cpio -ov > /backup/pacote.cpio
```

O comando acima empacotará todos os objetos da saída do comando "ls /etc". Para visualizar seu conteúdo:

```
1 # cpio -iv --list < /backup/pacote.cpio
```

Vamos renomear o diretório “/etc” para “/etc.old” :

```
# mv /etc /etc.old
```

Restaure o backup do /etc:

```
# cpio -iv < /backup/pacote.cpio
# ls /etc
```

Adicione arquivos ao pacote.cpio já criado:

```
# find /boot | cpio -ov -A -F /backup/pacote.cpio
```

Para visualizar seu conteúdo:

```
# cpio -iv --list < /backup/pacote.cpio
```

Adicione um usuário chamado inexistente:

```
# useradd inexistente
```

Verifique que ele foi criado no arquivo /etc/passwd:

```
# tail -n1 /etc/passwd
```

O comando cpio somente volta os arquivos, caso ele não exista ou ele seja mais recente que o atual:

```
1 # cpio -iv < /backup/pacote.cpio
```

Verifique que os arquivos não foram alterados, pois as datas são mais atuais ou as mesmas dos arquivos no pacote:

```
1 # tail -n1 /etc/passwd
```

Para forçar a restauração faça:

```
1 # cpio -iuv < /backup/pacote.cpio
```

Verifique que não existe mais o usuário “inexistente”:

```
1 # tail -n1 /etc/passwd
```

Remova o arquivo /etc/passwd:

```
1 # rm -f /etc/passwd
```

Restaure apenas o arquivo /etc/passwd:

```
1 # cpio -ivF /backup/pacote.cpio /etc/passwd
```

2.3 O empacotador tar

O que os compactadores como gzip e bzip2 não conseguem fazer, o tar (Tape Archives) faz. Ele é um aplicativo capaz de armazenar vários arquivos em um só. Porém, não é capaz de compactar os arquivos armazenados. Como é possível notar, o tar serve de complemento para os compactadores e vice-versa. Por isso, foi criado um parâmetro no tar para que ambos os programas possam trabalhar juntos. Assim, o tar "junta"os arquivos em um só e este arquivo, por sua vez, é então compactado por um dos compactadores suportados pelo tar.

O tar também consegue gravar a propriedade e as permissões dos arquivos. Ainda, consegue manter a estrutura de diretórios original (se houve compactação com diretórios), assim como as ligações diretas e simbólicas.

A sintaxe do TAR é:

```
1 tar [parâmetros] [-f arquivo] [-C diretório] [arquivos...].
```

Abaixo, segue a lista dos principais parâmetros:

- **-c ->** cria um novo arquivo tar;
- **-p ->** mantém as permissões originais do(s) arquivo(s);
- **-r ->** acrescenta arquivos a um arquivo tar;
- **-t ->** exibe o conteúdo de um arquivo tar;
- **-v ->** exibe detalhes da operação;
- **-x ->** extrai arquivos de um arquivo tar;

- **-z ->** comprime ou extrai arquivos tar resultante com o gzip;
- **-j ->** comprime ou extrai arquivos tar resultante com o bzip2;
- **-f ->** especifica o arquivo tar a ser usado;
- **-C ->** troca de diretório, para local de armazenamento ou restauração de dados.

Vamos empacotar o diretório “/etc” e “/usr”:

```
1 # tar -cvf /backup/etc.tar /etc
2 # tar -cvf /backup/usr.tar /usr
```

Verifique o tamanho do pacote “usr.tar”:

```
1 # du -sh /backup/usr.tar
```

Podemos utilizar parâmetros para reduzir o tamanho do pacote, através de compactadores como gzip e bzip2. Vamos observar o tempo com o comando time e o tamanho dos dois compactadores para vermos suas vantagens e desvantagens.

Empacotando e compactando com gzip o diretório “/usr”:

```
1 # time tar -zcvf /backup/usr.tar.gz /usr
```

Empacotando e compactando com bzip2 o diretório “/usr”:

```
1 # time tar -jcvf /backup/usr.tar.bz2 /usr
```

Podemos observar que o compactador gzip é mais rápido que o bzip2, mas qual deles fez a melhor compactação:

```
1 # du -sh /backup/usr*
```

O bzip2 faz a melhor compactação, mas em compensação leva um tempo maior que o gzip. Para visualizar o conteúdo dos pacotes tar faça:

```
1 # tar -tf /backup/usr.tar.gz
2 # tar -tf /backup/usr.tar.bz2
3 # tar -tf /backup/usr.tar
4 # tar -tf /backup/etc.tar
```

Para adicionar arquivos ao pacote já criado utilize o parâmetro "-r", mas somente é possível em pacotes que ainda não foram compactados: Crie um arquivo chamado "aaaaaaaaaaaaaaaaaa" em /etc:

```
1 # touch /etc/aaaaaaaaaaaaaaaaaa
```

Adicione ao tar criado:

```
1 # tar -rf /backup/etc.tar /etc/aaaaaaaaaaaaaaaaaa
```

Visualize:

```
1 # tar -tf /backup/etc.tar
```

Vamos acessar o diretório /backup e descompactar o "/usr" feito com gzip:

```
1 # cd /backup
2 # tar -zxvf usr.tar.gz
```

Verifique que o pacote compactado foi descompactado no diretório atual e não na raiz:

```
1 # ls
```

Para determinar qual vai ser o ponto inicial para descompactar o pacote utilize o parâmetro “-C”. Descompacte o pacote feito com bzip2 no diretório /mnt:

```
1 # tar -jxvf usr.tar.bz2 -C /mnt
```

Verifique o diretório /mnt:

```
1 # ls /mnt
```

Agora delete o diretório /etc:

```
1 # rm -rf /etc
```

Volte o backup feito com o tar a partir do diretório “/”:

```
1 # tar xvf etc.tar -C /
```

Verifique:

```
1 # ls /etc
```

2.4 Compactadores GZIP, BZIP2

Compressão de dados é o processo de codificar a informação de forma que seja possível armazená-la em um número menor de "bits". Por exemplo, se definíssemos que a palavra "compressão" passaria a ser abreviada por "comp", estaríamos diminuindo o número de "bits" necessários para armazenar esta apostila.

Entretanto, para que você pudesse entender o que "comp" significa seria necessário estar ciente dessa convenção ou seja, do algoritmo de compressão.

Há dois tipos básicos de compressão, aquele em que não há perdas de informações e aquele em que elas ocorrem. Obviamente quando o assunto é "backup" de informações vitais, devemos utilizar algoritmos sem perdas. Já em arquivos de imagens, vídeos e áudio, há casos que podemos nos dar ao luxo de perdas de informações em detrimento da qualidade, que em geral é praticamente imperceptível para os não especialistas da área.

Os principais programas de compressão que utilizaremos são o "bzip2" e "gzip" . O "bzip2" utiliza os algoritmos "Burrows-Wheeler transform" e "Huffman coding"; já o "gzip" utiliza os algoritmos "LZ77" e "Huffman coding". Todos esses algoritmos fazem parte do grupo dos algoritmos que não ocasionam perdas de dados.

A forma de utilização desses comandos é bastante simples. Para o "gzip", "bzip2", basta fornecer o arquivo de entrada que a compressão se dará no próprio arquivo. Eis uma diferença entre o "tar" e esses programas, ele recebe dois argumentos, os arquivos de entrada e o arquivo de saída, ou seja, aqueles a serem empacotados e comprimidos.

Verifique que não é possível compactar um diretório sem empacotá-lo antes. Tente

com o “gzip” e com o “bzip2”:

```
1 # gzip etc
2 # bzip2 etc
```

Para determinarmos qual o melhor compactador vamos analisar dois tipos de arquivos: texto puro e binário. Para isso vamos criá-los.

Crie dois arquivos de texto puro. Abra o arquivo “texto1” no editor “vim” e insira uma linha contento os números de 0 a 9:

```
1 # vim texto1
2 0123456789
```

Ainda dentro do “vim”, copie essa linha e cole “250.000” vezes e salve o arquivo:

```
1 <ESC>
2 yy
3 250000p
4 :x!
```

Crie uma cópia deste arquivo chamando-a de “texto2”:

```
1 # cp texto1 texto2
```

Crie um par de arquivos binários para nossos testes. Utilizaremos como base, o programa “aptitude”:

```
1 # cp /usr/bin/aptitude bin1
```

Duplique esse arquivo:

```
1 # cp bin1 bin2
```

Verifique que foram criados quatro arquivos com tamanhos parecidos, aproximadamente 2.4MB, sendo dois deles binários e dois texto puro:

```
1 # ls -lh bin* texto*
```

2.4.1 Gzip e Bzip2 com Arquivos de Texto

Utilize a tabela “tab:comparacao1” para anotar os resultados obtidos nos testes com “gzip” e “bzip” em arquivos de texto puro:

Tipo do Arquivo: texto puro **Tamanho Original:**

	GZIP	*BZIP2*
Tamanho final		
Tempo para compactar		
Tempo para descompactar		

 Vamos iniciar os testes com os arquivos texto.

Determine o intervalo de tempo que leva para comprimir o arquivo “texto1” com “gzip”:

```
1 # time gzip texto1
```

Determine o tamanho final do arquivo “texto1” após ser comprimido com “gzip”:

```
1 # ls -lh texto1.gz
```

Determine o intervalo de tempo que leva para descomprimir o arquivo “texto1.gz”:

```
1  # time gunzip texto1.gz
```

Vamos repetir os procedimentos utilizando o “bzip2”: Determine o intervalo de tempo que leva para comprimir o arquivo “texto2” com “bzip2”:

```
1 # time bzip2 texto2
```

Determine o tamanho final do arquivo “texto2” após ser comprimido com “bzip2”:

```
1 # ls -lh texto2.bz2
```

Determine o intervalo de tempo que leva para descomprimir o arquivo “texto2.bz2”:

```
1 # time bunzip2 texto2.bz2
```

2.4.2 Gzip e Bzip2 com Arquivos Binários

Utilize a tabela "tab:comparacao2" para anotar os resultados obtidos nos testes com "gzip" e "bzip" em arquivos binários:

Tipo do Arquivo: binário **Tamanho Original:**

	GZIP	*BZIP2*
Tamanho final		
Tempo para compactar		
Tempo para descompactar		

Determine o intervalo de tempo que leva para comprimir o arquivo "bin1" com "gzip":

```
1 # time gzip bin1
```

Determine o tamanho final do arquivo "bin1" após ser comprimido com "gzip":

```
1 # ls -lh bin1.gz
```

Determine o intervalo de tempo que leva para descomprimir o arquivo "bin1.gz":

```
1  # time gunzip bin1.gz
```

Vamos repetir os procedimentos utilizando o "bzip2": Determine o intervalo de tempo que leva para comprimir o arquivo "bin2" com "bzip2":

```
1 # time bzip2 bin2
```

Determine o tamanho final do arquivo "bin2" após ser comprimido com "bzip2":

```
1 # ls -lh bin2.bz2
```

Determine o intervalo de tempo que leva para descomprimir o arquivo “bin2.bz2”:

```
1 # time bunzip2 bin2.bz2
```

2.5 Comando dd

O comando “dd” tem a capacidade de copiar “bit a bit”. Segue um exemplo de seu uso: FAZER PARTIÇÃO MENOR E COPIAR

```
1 # dd if=/dev/sda3 of=/dev/sda11
```

O comando acima efetuará a clonagem da partição “sda3”, para a partição “sda11”.

Cuidado com o comando “dd”, qualquer falta de atenção pode danificar o sistema, de forma irrecuperável.

Onde:

if=**/dev/sda3**

O nome do arquivo de entrada.

of=**/dev/sda11**

4451

Linux System Administration

www.4linux.com.br

Conteúdo

Backup com ferramentas XFS

3.1 Introdução Teórica

O XFS é um sistema de arquivos de alta performance com suporte a journaling, que teve origem na plataforma IRIX da SGI. É completamente multi-processo, e pode suportar grandes sistemas de arquivos com atributos estendidos, tamanho de blocos variável. O XFS é baseado em extents e utiliza bem o uso de Btrees (diretórios, extensões, e espaço livre) para ajudar no ganho de performance e escalabilidade.

3.2 Gerenciando backup em partições XFS

Existem diversas ferramentas para gerenciar partições do tipo XFS, em nossa pratica vamos manipular a aplicação do sistema de arquivos, backup e restore no sistema de arquivos XFS. Antes de usar os comandos prepare sua infra adicionando um novo disco para o backup. Um novo disco sera usado em /dev/sdb para as tarefas, sendo que a primeira partição /dev/sdb1 deve conter 2GB e a segunda 6GB:

Aplicando sistema de arquivos XFS

O comando mkfs.xfs é usado para aplicar sistema de arquivos XFS em uma partição. O comando também pode ser usado com a flag -t como em “mkfs -t xfs”.

```
1 # mkfs.xfs /dev/sdb1
2 Ou
3 # mkfs -t xfs /dev/sdb2
```

Como a partição já possui o sistema de arquivos XFS, crie o ponto de montagem, monte a partição e faça a cópia de novos arquivos.

```
1 # mkdir /media/xfs
2 # mount -t xfs /dev/sdb1 /media/xfs
3 # cp -R /var /media/xfs
```

3.3 Gerar backup com xfsdump

O comando xfsdump é utilizado para fazer backup de arquivos com seus atributos em um sistema de arquivos. O xfsdump examina os arquivos e determina quais precisam ser salvos (backup), e copia esses arquivos para um disco especificado, como fita magnética ou outra média de armazenamento.

A ferramenta usa diretivas específicas do XFS para otimização, e também sabe como salvar os atributos extensos do XFS. Os backups criados pelo xfsdump são do tipo "endian safe"e assim podem ser transferidos entre máquinas Linux de diferentes arquiteturas e também entre máquinas IRIX.

Crie um local para nosso backup em /media usando a segunda partição de 6GB

```
1 # mkdir /media/backup
2 # mount -t xfs /dev/sdb2 /media/backup
```

Use o comando xfsdump para fazer o backup da partição /dev/sdb1

```
1  # xfsdump -l 0  -p 30 -f /media/backup/backup.0.dump  /media/xfs
2
3  xfsdump: using file dump (drive_simple) strategy
4  xfsdump: version 3.0.4 (dump format 3.0) - Running single-threaded
5
6  ==================== dump label dialog ====================
7
8  please enter label for this dump session (timeout in 300 sec)
9   -> backup (Digite o nome do rotulo e tecle Enter)
10
11 session label entered: "backup"
12
13 ----------------------- end dialog -----------------------
14
15 xfsdump: level 0 dump of debian:/media/xfs
16 xfsdump: dump date: Thu Apr 26 16:05:12 2012
17 xfsdump: session id: ce153353-0d33-4fc5-aa88-10274eeaff4b
18 xfsdump: session label: "backup"
19 xfsdump: ino map phase 1: constructing initial dump list
20 xfsdump: ino map phase 2: skipping (no pruning necessary)
21 xfsdump: ino map phase 3: skipping (only one dump stream)
22 xfsdump: ino map construction complete
23 xfsdump: estimated dump size: 171053504 bytes
24
25 ==================== dump label dialog ====================
26
27 please enter label for media in drive 0 (timeout in 300 sec)
28  -> quit (Digite o comando para sair do prompt do xfsdump)
29
30 ----------------------- end dialog -----------------------
31
32 xfsdump: creating dump session media file 0 (media 0, file 0)
33 xfsdump: dumping ino map
34 xfsdump: dumping directories
35 xfsdump: dumping non-directory files
36 xfsdump: status at 16:05:46: 1/3842 files dumped, 0,0% data dumped,
```

```
   34 seconds elapsed
37 xfsdump: ending media file
38 xfsdump: media file size 160228640 bytes
39 xfsdump: dump size (non-dir files) : 158883080 bytes
40 xfsdump: dump complete: 38 seconds elapsed
41 xfsdump: Dump Status: SUCCESS
```

Opções utlizadas:

- **-l:** Especifica um nível de dump de 0 a 9. O nível de dump determina o que sera armazenados no backup. Um dump de de nível 0 é absoluto onde todos os arquivos são armazenados. Um nível de dump 1 a 9 é referido como um dump incremental. Apenas os arquivos que foram alterados desde o dump de base (nivel 0) são despejados.

- **-p:** Faz com que relatórios de progresso sejam impressos no intervalo especificado (intervalo é dada em segundos). O relatório indica quantos arquivos foram despejadas, o número total de arquivos para descarregar, a porcentagem de dados, objeto de dumping, e o tempo decorrido.

- **-f:** Especifica um destino de despejo. Um destino pode ser o caminho de um dispositivo (como uma unidade de fita), um arquivo regular ou uma unidade de fita remota.

Onde foi parar o backup da partição /dev/sdb1?

Como resultado do comando foi criado um arquivo com a extensão .dump no diretório /media/backup. Através deste arquivo podemos restaurar o backup completo da partição /dev/sdb1.

Como incrementar o backup com apenas arquivos alterados?

Primeiro faça uma alteração na partição gravando um novo arquivo ou alterando arquivos já gravados.

```
# echo "Esta é uma nova alteração" >> /media/xfs/var/log/mail.log
```

Use o comando xfsdump para criar um backup do conteúdo alterado (backup incremental)

```
# xfsdump -l 1 -p 30 -f /media/backup/backup.1.dump  /media/xfs

xfsdump: using file dump (drive_simple) strategy
xfsdump: version 3.0.4 (dump format 3.0) - Running single-threaded
xfsdump: level 1 incremental dump of debian:/media/xfs based on
    level 0 dump begun
xfsdump: dump date: Thu Apr 26 16:40:17 2012
xfsdump: session id: ff2f5ed8-977c-417e-8538-40513baf6939
xfsdump: session label: "backup"
xfsdump: ino map phase 1: constructing initial dump list
xfsdump: ino map phase 2: pruning unneeded subtrees
xfsdump: ino map phase 3: skipping (only one dump stream)
xfsdump: ino map construction complete
xfsdump: estimated dump size: 38464 bytes

 ==================== dump label dialog ====================
please enter label for media in drive 0 (timeout in 300 sec)
 -> backup (Digite o nome do rotulo e tecle Enter)
media label entered: "backup"

 ----------------------- end dialog -----------------------
xfsdump: creating dump session media file 0 (media 0, file 0)
xfsdump: dumping ino map
xfsdump: dumping directories
xfsdump: dumping non-directory files
xfsdump: ending media file
xfsdump: media file size 33968 bytes
xfsdump: dump size (non-dir files) : 10304 bytes
xfsdump: Dump Status: SUCCESS
```

Como resultado do comando foi criado um novo arquivo com a extensão .dump no diretório /media/backup (backup.1.dump). Neste arquivo temos o incremento da partição (o que tem de novo). Compare o tamanho dos dois arquivos:

```
1 # cd /media/backup
2 # ls -lh
3 -rw-r--r-- 1 root root 153M Abr 26 16:29 backup.0.dump
4 -rw-r--r-- 1 root root  34K Abr 26 16:40 backup.1.dump
```

3.4 Restaurando backup com xfsrestore

O comando xfsrestore executa a função reversa do xfsdump; podendo restaurar uma cópia de segurança completa de um sistema de arquivos. Backups incrementais subsequentes podem ser colocados depois 'em cima' do backup completo. Arquivos únicos e subdiretórios podem ser restaurados a partir de backups completos ou parciais.

Primeiro crie um novo local para armazenar o backup restaurado, e use o comando xfsrestore para restaurar o backup incremental:

```
1 # mkdir /backup
2 # xfsrestore  -f /media/backup/backup.1.dump /backup/
3
4 xfsrestore: using file dump (drive_simple) strategy
5 xfsrestore: version 3.0.4 (dump format 3.0) - Running single-
    threaded
6 xfsrestore: searching media for dump
7 xfsrestore: examining media file 0
8 xfsrestore: dump description:
9 xfsrestore: hostname: debian
10 xfsrestore: mount point: /media/xfs
```

```
11 xfsrestore: volume: /dev/sdb1
12 xfsrestore: session time: Thu Apr 26 16:40:17 2012
13 xfsrestore: level: 1
14 xfsrestore: session label: "backup"
15 xfsrestore: media label: "backup"
16 xfsrestore: file system id: 3eca0743-cfa5-45b4-8376-c892e3793511
17 xfsrestore: session id: ff2f5ed8-977c-417e-8538-40513baf6939
18 xfsrestore: media id: ddf5382d-3770-4aef-861f-e390edf46cd5
19 xfsrestore: using online session inventory
20 xfsrestore: searching media for directory dump
21 xfsrestore: reading directories
22 xfsrestore: 3 directories and 54 entries processed
23 xfsrestore: directory post-processing
24 xfsrestore: restoring non-directory files
25 xfsrestore: restore complete: 0 seconds elapsed
26 xfsrestore: Restore Status: SUCCESS
```

Toda a estrutura de diretórios com o arquivo auth.log foi restaurado. Liste o conteudo do diretório /backup e comprove a restauração.

```
1 # ls -l /backup/var/log/auth.log
2 -rw-r----- 1 root root 9422 Abr 26 16:38 /backup/var/log/auth.log
```

Caso precise de uma restauração completa do /dev/sdb1, use o comando:

```
1 # xfsrestore  -f /media/backup/backup.0.dump /media/xfs
```

3.5 Criar e Restaurar Backup Remoto

A pratica é bem parecida com o procedimento local mudando apenas, comandos adicionados para a conexão remota e a compactação dos arquivos. Em nossa infra

vamos precisa de 2 maquinas com SSH.

- **Maquina 1:** Debian 6 com IP 192.168.200.1 (pacote xfsdump instalado)
- **Maquina 2:** CentOS 6 com IP 192.168.200.2 (pacote xfsdump instalado)

Backup Completo via SSH

Na maquina Debian 6 use o comando xfdump para criar um backup completo do diretório /media/xfs.

```
1 # xfsdump -l 0 -L backup  - /media/xfs | gzip | ssh root@192
    .168.200.2 dd of=/backup/backup$(date +%d-%m-%Y).gz
```

Descrição das opções utilizadas:

- **xfsdump -l 0 -L backup - /media/xfs | :** Comando usado para criar o backup completo (-l 0) do diretório /media/xfs com o label "backup";
- **gzip | :** O resultado do comando xfsdump sera compactado atraves deste comando;
- **ssh root@192.168.200.2 :** Envia o backup compactado para a maquina remota;
- **dd of=/backup/backup$(date +%d-%m-%Y).gz :** Na maquina remota o backup compactado sera gravado em um arquivo (dd of) com a data atual em /backup.

Através da linha de comando em nosso exemplo, o backup foi criado de forma remota na maquina CentOS no diretório /backup. Não esqueça de criar este diretório na maquina CentOS.

Restore completo via SSH

Para restaurar use o comando xfsrestore na maquina Debian apontando o diretório de destino:

```
1 # ssh root@192.168.200.2 "dd if=/backup/backup30-04-2012.gz" |
     gunzip -c | xfsrestore - /mnt/
```

Descrição das opções utilizadas:

- **ssh root@192.168.200.2 :** Recebe o backup compactado da maquina remota;
- **dd if=/backup/backup30-04-2012.gz | :** Na maquina remota o backup compactado sera lido através de um arquivo (dd if);
- **gunzip -c | :** O resultado do comando anterior sera descompactado atraves deste comando;
- **xfsrestore - /mnt/backup/ :** Comando usado para restaurar o backup da maquina remota no diretório /opt.

Backup incremental via SSH

Na maquina Debian altere um arquivo no diretório /media/xfs e crie um backup incremental (-l 1) na maquina CentOS:

```
1 # echo "Novo conteudo" >> /media/xfs/var/log/auth.log
2 # xfsdump -l 1 -L backup  - /media/xfs | gzip | ssh root@192
     .168.200.2 dd of=/backup/backup-incremental01-$(date +%d-%m-%Y).
     gz
```

Restore incremental via SSH

Para restaurar apenas o que foi incrementado no backup completo:

```
1 # ssh root@192.168.200.2 "dd if=/backup/backup-incremental01
    -30-04-2012.gz" | gunzip -c | xfsrestore - /opt
```

Dica: Para não precisar informar a senha crie uma configuração com chaves no SSH entre as maquinas Debian e CentOS!

Agendamento de Tarefas

3.6 Introdução Teórica

A **“crontab”** é utilizada para agendar comandos que serão executados periodicamente, ao contrário do comando “at”, que executa comandos pontualmente. Há dois tipos de “crontab”: a de **usuários** e a do **sistema.** Ambas são arquivos que contêm tabelas com informação de quando o comando especificado deve ser executado, sendo que cada linha corresponde a um único **agendamento.**

A “crontab” é gerenciada pelo “daemon crond”, que a cada um minuto verifica se há algum agendamento que deve ser executado e, se houver, executa-o.

A “crontab” dos usuários pode ser acessada pelo comando:

```
# crontab [-e|-r|-l]
```

A tabela fica armazenada em arquivos com o nome do usuário dono da tabela. Já a “crontab” do sistema é encontrada no arquivo **“/etc/crontab”** e já possui agendamentos para realizar as tarefas que se encontram nos diretórios **“/etc/cron.[hourly|daily|weekly|month**
Sendo que o programa chamado “run-parts” é quem executa os referidos agendamentos.

O formato das “crontabs” dos usuários e do sistema são quase iguais. A exceção é que a “crontab” do sistema possui um campo a mais, como pode ser visto a seguir:

```
1 crontab (usuários)
2 # minuto hora dia mês diaDaSemana comando
3
4 crontab (sistema)
5 # minuto hora dia mês diaDaSemana USUÀRIO comando
```

A única diferença entre as duas “crontabs” é que na do sistema há um campo para especificar qual é o usuário que irá executar o comando agendado.

Além disso cada campo possui um conjunto de valores válidos, sendo eles:

minuto: varia de 0-59;

hora: varia de 0-23;

dia: varia de 1-31;

mês: varia de 1-12;

diaDaSemana: varia de 0-7, sendo:

0 ou 7 – domingo 1 - segunda-feira 2 - terça-feira 3 - quarta-feira 4 - quinta-feira 5 - sexta-feira 6 - sábado

- **usuário:** um usuário válido no sistema;
- **comando**: o “path” completo para o comando.

Podemos controlar quais usuários podem acessar ou não o “cron”. Para isso basta criar um dos arquivos: “/etc/cron.allow” ou “/etc/cron.deny”. A mesma dica é válida para o comando “at”: “/etc/at.allow” ou “at.deny”.

Considerando o formato já listado, podemos realizar agendamentos utilizando alguns operadores que facilitam o trabalho. São eles:

- **vírgula (,) ->** especifica uma lista de valores, por exemplo: “1,3,4,7,8”;
- **hifen (-) ->** especifica um intervalo de valores, por exemplo: 1-15 (de 1 a 15);
- **asterisco (*) ->** especifica todos os valores possíveis;
- **barra (/) ->** especifica “pulos” de valores, por exemplo: se no campo hora utilizarmos “*/3” o comando será executado às “0,3,6,9,12,15,18,21” horas;

3.6.1 Agendamento de Tarefas com AT

O comando "at"pode agendar tarefas de forma semelhante ao cron, e é integrado à interface de linha de comando do Linux. É muito eficiente se aplicado no agendamento de tarefas que sejam disparadas somente uma vez. O at permite o controle dos usuários que podem agendar comandos através dos arquivos /etc/at.allow e /etc/at.deny. Estes arquivos são organizados no formato de um usuário por linha. Durante o agendamento é verificado primeiro o arquivo /etc/at.allow (listando quem pode executar o comando) e depois /etc/at.deny. Caso eles não existam, o agendamento de comando é permitido a todos os usuários.

Verifique se a data e a hora do sistema estão corretas:

```
1 # date
```

 Após essa verificação podemos começar a realizar agendamentos.

Agende para 10 minutos no futuro um backup do diretório “/etc”, colocando seu backup no diretório /backup.

Agende a tarefa de backup:

```
1 # at HH:mm MM/DD/YYYY
2 at> tar zcvf /backup/backup-etc.tar.gz /etc/
3 at> (Ctrl + d)
```

Agendada esta tarefa, confirme-a listando todos os agendamentos pendentes:

```
1 # atq
```

Vamos explorar o diretório onde ficam os agendamentos:

 # cd /var/spool/cron/atjobs # ls -la

 # cd /var/spool/at # ls -la

Mostre o conteúdo dos arquivos contidos nesse diretório:

```
# cat (agendamento)
```

 Perceba que no agendamento, temos nossas variáveis e o comando.

Vamos realizar outro agendamento qualquer, para executar em 15 minutos, para que possamos aprender como apagá-lo:

```
# at HH:mm MM/DD/YYYY
at> echo "Teste" > /tmp/at.out
at> ^d
```

Liste os agendamentos correntes e verifique que um novo arquivo foi criado no diretório de “spool” do “at”.

```
# cd /var/spool/cron/atjobs
# ls -la
```

Agendada esta tarefa, confirme-a listando todos os agendamentos pendentes:

```
# atq
```

1) Remova o último agendamento:

```
# atrm [número_agendamento]
```

Liste os agendamentos ativos e liste o conteúdo do diretório de “spool” do “at” e veja que o “job” foi removido:

```
# atq
# ls /var/spool/cron/atjobs
```

Verifique o backup que estava agendado pelo comando at:

```
# ls /backup
```

Todos usuários comuns podem utilizar o comando at, por padrão somente vem criado o arquivo “/etc/at.deny”, neste arquivo são configurados os usuários que não podem utilizar o agendador de tarefas at, caso queira bloquear o uso de alguns usuários específicos adicione-os neste arquivo, sendo um usuário por linha, se quiser bloquear o uso de todos, crie o arquivo “/etc/at.allow” em branco.

Caso queira habilitar o uso do agendador de tarefas at para apenas alguns usuários específicos, crie o arquivo “/etc/at.allow” e coloque um nome por linha.

Bloqueie o uso do agendador de tarefas at para usuários comuns:

```
# touch /etc/at.allow
```

Teste o bloqueio com o usuário mandark:

```
$ at 12:00 01/01/2014
```

3.6.2 Agendando Tarefas com o CRON

Cuidados especiais com scripts

Utilize nos comandos do script e no agendador, sempre o (path) caminho completo do aplicativo a ser executado, exemplo para o comando tar, use /bin/tar, também na linha de comando que inserir no cron use o caminho completo para o script, por exemplo, executar um script que está em /home/zago, use a linha: /home/zago/nome-do-script e não somente nome-do-script.

Muito cuidado com scripts, o comodismo pode cair no esquecimento e não atualizar o script de backup quando incluir novos serviços, diretórios ou usuários, monitore constantemente, teste e avalie o que está sendo feito.

Tenha os seguintes cuidados quando elaborar scripts para execução pelo cron.

Nestes scripts não pode conter comandos que requer interação com o usuário, tais como pedir senha para completar a conexão de um ftp, nestes casos deve ser colocado todas as instruções dentro do script de maneira que possa completar a conexão passando o login e senha.

Comandos que requerem confirmação para execução, por exemplo, apagar arquivos, o rm pede confirmação, mas com o parâmetro -rf não pede, então seria assim: rm -rf <arquivo, diretório ou /caminho/o que deve apagar>

Não deve ter nenhum comando que peça confirmação ou qualquer interação com o usuário.

Fique atento às permissões, quando possível agende como root para executar o script, use o "sudo"para dar permissões de execução em programas que requer poderes de root na execução, acesso a diretórios de backup e etc....

Espaço em disco quando baixar arquivos, backup ..., comandos de parar serviços ou manipular arquivos em uso. Revise periodicamente scripts que requerem atualiza-

ção, tais como backup de dados dos usuários, incluir novos usuários...

O cron limita a busca nos diretórios /bin e /usr/bin, portanto indique o caminho completo do programa ou script, ou melhor indique sempre, mesmo que estejam nestes diretórios.

Para agendar as tarefas usamos o comando crontab com a sintaxe descrita abaixo:

```
1 # crontab [-u usuário] { -e | -l | -r }
```

u -> permite que o superusuário agende tarefas para outros usuários, pois o comando su pode atrapalhar o crontab. Um usuário comum não precisa usar essa opção para especificar ele próprio.

e -> edita o arquivo de tarefas agendadas pelo usuário. A formatação desse arquivo será descrita mais adiante.

l -> lista o arquivo de tarefas agendadas pelo usuário.

r -> apaga o arquivo de tarefas agendadas pelo usuário.

Basicamente, para agendarmos uma tarefa deveremos editar o nosso arquivo "agenda" com o comando:

```
1 #  crontab -e
```

O arquivo agenda tem as seguintes características: as linhas em branco, espaços iniciais e tabs são ignorados. As linhas cujo primeiro caractere não-branco for um "#"são comentários, e são ignorados. Uma linha ativa em um arquivo agenda é uma definição de ambiente ou um comando do cron.

Definição de ambiente:

nome=valor -> A string valor pode ser colocada entre aspas (simples ou duplas, mas correspondentes) para preservar espaços iniciais ou finais.

Várias variáveis de ambiente são definidas automaticamente pelo servidor cron. SHELL é definida como /bin/sh, LOGNAME e HOME são definidos a partir da linha do /etc/passwd referente ao usuário que agendou a tarefa. HOME e SHELL podem ser modificadas, mas LOGNAME não.

O formato de um comando do cron é em grande parte o padrão V7. Cada linha tem cinco campos de hora e data, seguidos por um comando. Os comandos são executados pelo servidor cron quando os campos minuto, hora, e mês correspondem à hora atual, e quando pelo menos um dos campos de dia (dia do mês, ou dia da semana) correspondem ao dia atual.

Entre na “crontab” do usuário para editá-la:

```
1 # crontab -e
```

Para entendermos a diferença entre os campos “dia do mês” e “dia da semana”, vamos agendar uma tarefa no crontab para escrever a data no terminal 2:

```
1 # minuto hora dia_do_mes mês   dia_da_semana comando
2    *      *     13        09        6         /bin/date > /dev/tty2
```

No comando acima foi feito um agendamento para ser executado, todos os minutos de todas as horas, no dia 13 de setembro e todos os sábados de setembro, ou seja, os campos “dia do mês” e “dia da semana” trabalham de forma separada. Vamos mudar nossa data e hora para verificarmos:

```
1 # date 091323582013
```

Mudamos a data para o dia 13 de setembro de 2013, uma sexta-feira. Visualize a data:

```
# cal 2013
```

Visualize no terminal 2 pela data, que, é executado o comando "hoje", dia 13 de setembro de 2013 e que "amanhã", sábado ele também é executado.

Visualize os agendamentos feitos pelo o usuário.

```
# crontab -l
```

Onde ficam armazenados os agendamentos feitos pelos usuários com o "crontab -e"?

 # cd /var/spool/cron/crontabs # ls

 # cd /var/spool/cron # ls

Não apague ou edite o seu agendamento dentro desse diretório, use os comandos para fazer isso.

Após verificar que os agendamentos foram efetuados corretamente, apague todos os agendamentos do usuário.

```
# crontab -r
```

Para apagar somente um agendamento do usuário, use o “crontab -e” e retire a linha desejada.

Agora que aprendemos a utilizar a “crontab” do usuário podemos usar a “crontab” do sistema que opera praticamente da mesma forma, apenas tem um campo a mais, o usuário que executará o “script”. Seu arquivo de configuração é o /etc/crontab. **Debian:**

```
# cat /etc/crontab
SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

#m h dom mon dow user command
17 * * * * root    cd / && run-parts --report /etc/cron.hourly
25 6 * * * root test -x /usr/sbin/anacron || ( cd / && run-parts --
   report /etc/cron.daily )
47 6   * * 7 root test -x /usr/sbin/anacron || ( cd / && run-parts
   --report /etc/cron.weekly )
52 6 1 * * root test -x /usr/sbin/anacron || ( cd / && run-parts --
   report /etc/cron.monthly )
```

CentOS:

```
# cat /etc/crontab
SHELL=/bin/sh
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root
HOME=/

01  *  *  *  *  root    run-parts /etc/cron.hourly
02  4  *  *  *  root    run-parts /etc/cron.daily
```

```
9  22  4  *  *  0  root    run-parts /etc/cron.weekly
10 42  4  1  *  *  root    run-parts /etc/cron.monthly
```

O programa *"run-parts"* executa todos os scripts executáveis dentro de um certo diretório. Então com essas linhas, temos diretórios programados para executar programas de hora em hora, diariamente, semanalmente ou mensalmente. Abaixo a tabela:

Ainda dentro do diretório /etc, temos quatro agendamentos pré-definidos:

cron.hourly, cron.daily, cron.weekly e cron.montly Onde:

cron.hourly: de hora em hora

cron.daily: de dia em dia

cron.weekly: de semana em semana

cron.montly: de mês em mês

3.6.3 Restringindo o uso do crontab

Os arquivos "/etc/cron.allow" e "/etc/cron.deny" são usados para restringir acesso ao cron. O formato de ambos arquivos de controle de acesso consiste em um nome de usuário por linha. Espaços em branco não são permitidos em nenhum destes arquivos. O daemon do cron não precisa ser reiniciado se os arquivos de controle de acesso forem modificados. Os arquivos de controle de acesso são lidos a cada vez que o usuário tentar adicionar ou apagar uma tarefa do cron.

O usuário root pode usar o cron sempre, independentemente dos nomes de usuário listados nos arquivos de controle de acesso.

Se o arquivo /etc/cron.allow existe, somente os usuários listados neste poderão usar o cron, e então o arquivo cron.deny será ignorado.

Se o arquivo cron.allow não existe, os usuários listados no cron.deny não poderão usar o cron.

4451

Linux System Administration

www.4linux.com.br

Conteúdo

Automatização de Tarefas com Shell Script I

4.1 Introdução Teórica

Tarefas administrativas são, muitas vezes, longas e repetitivas. Podemos automatizar esses procedimentos através de “scripts”. Na verdade, os “scripts” podem nos auxiliar muito, numa vasta gama de atividades.

O que é um “script”? É uma sequência de instruções que são executadas toda vez que o mesmo é chamado.

Mas, qual a diferença entre um “script” e um programa, já que ambos são sequências de instruções?

Um “script”, é um programa não compilado. O processador da máquina só é capaz de executar programas binários, isto é, compilados especificamente para ele. Dessa forma, é necessário um programa que interprete esse “script”, em tempo de execução, para que o mesmo possa ser executado. No nosso caso, esse programa será uma “shell”, já que estamos falando de “shell scripts”.

Sendo uma linguagem de programação, a “Shell Script” possui uma série de estruturas de controle como “loops” e condicionais, mas que são estudadas apenas em cursos mais avançados. Estudando um exemplo

Vejamos o seguinte exemplo de “Shell Script”:

```
1 # vim shell.sh
2 #!/bin/bash
3 # Meu primeiro shell script
4 cd ~
5 clear
6 ls -alh
7 date
8 cd -
```

Este é um “script” bem simples. As linhas que começam pelo símbolo "cedilha"são comentários, ou seja, tudo que aparece depois do desse carácter é desprezado. Os comentários são muito importantes nos programas, pois são uma forma de documentá-los. Imagine se você tiver que fazer uma alteração num programa escrito a um ano antes. Será que você irá se lembrar de todas as estruturas e variáveis que utilizou? Provavelmente não. Se for outra pessoa quem tiver que efetuar essa mudança, a situação será pior ainda!

Mas a primeira linha, na qual aparece um comentário, possui uma característica um tanto estranha. Na verdade, a primeira linha de um “script”, indica qual será o interpretador daquele “script”. Em nosso exemplo será o programa “binbash”, uma “shell”. Se estivéssemos criando um “script” com a linguagem de programação “Perl”, a primeira linha seria algo como usrbinperl.

O “script”, propriamente dito, executa 4 comandos simples: Acessar o diretório do usuário corrente (cd); limpar a tela (clear); listar o conteúdo diretório corrente (ls -alh); imprimir a data (date) e voltar ao diretório original (cd -). Executando o script

Um programa ou “script” no GNU/Linux deve possuir permissão de execução. Supondo que nosso “script” denomina-se “shell.sh”, para podermos executá-lo, devemos executar o comando:

```
1 # chmod u+x shell.sh
```

E em seguida executar o script:

```
# ./shell.sh
```

Em algumas situações, pode ser necessário fornecer parâmetros para um “script”. Por exemplo, se ao invés de listar o conteúdo do diretório pessoal do usuário, quiséssemos que o “script” listasse o conteúdo de um diretório qualquer.

Supondo que esse novo “script” chama-se “script2.sh”, uma possível forma de utilização do “script” seria:

```
# ./script2.sh /etc
```

Para passar parâmetros para esse “script”, precisamos conhecer a função de algumas variáveis: “$1, $2”. Quando passamos algum parâmetro para o nosso “script”, esse parâmetro fica armazenado em uma variável específica. Por exemplo:

```
# ./script3.sh  parâmetro1 parâmetro2 parâmetro3
```

Para conseguirmos resgatar o valor desses parâmetro, precisamos chamar as variáveis “$1,$2 e $3”, por exemplo:

```
# vim  script3.sh
#!/bin/bash
#
#Esse script pega o valor dos parâmetros e imprimi na tela.
echo $1
echo $2
echo $3
# ./script3 42 the answer
```

4.2 Usando os números

Muitas vezes quando fazemos “scripts”, precisamos de uma função que faça o trabalho das operações matemáticas básicas como soma, divisão, multiplicação e subtração. Em “shell script” podemos usar o comando “expr” para realizá-las. Já a contagem de linhas é feita pelo comando “wc”. E o comando “cut” serve para “cortar” a saída no ponto especificado pelo separador.

Vamos ver esse exemplo: um “script” que deve dizer quantos usuários estão presentes, quantos grupos estão presentes e no final mostrar quantos objetos meu sistema tem, a soma dos usuários e dos grupos:

```
1 #!/bin/bash
2 #
3 echo "Aguarde ....."
4 sleep 3
5 G=`wc -l /etc/group | cut -d" " -f1`
6 U=`wc -l /etc/passwd | cut -d" " -f1`
7 echo "O sistema possui $U usuários."
8 echo "O sistema possui $G grupos."
9 echo "O sistema possui `expr $G + $U` objetos."
```

4.3 Usando a estrutura “se”

Até o presente momento, fizemos “scripts” que não possuem escolhas, ou seja, comandos de execução em linha de comando em série, utilizando nossa lógica para fazer com que aquele “script” seja executado corretamente. Mas se qualquer coisa acontecer no meio do caminho, não temos a oportunidade de trabalhar com as famosas “exceptions”.

As “exceptions”, também conhecidas como exceções, servem para ajudar quando o

resultado de alguma parte do “script” pode ter vários rumos. Usando a condição “se”, é possível testar o resultado de uma condicional.

Por exemplo:

a=1 b=2 SE $b >$a ENTÃO IMPRIMA $bSENOIMPRIMA$a FIMSE

A variável “$?“

A variável interrogação é conhecida por testar o valor de retorno de qualquer comando quando mostrada após sua execução. Com ela podemos verificar se o programa foi executado com sucesso ou não. Para isso basta saber que essa variável tem dois retornos principais.

```
1 # pwd
2 # echo $?
3 # 0
```

Quando o resultado dessa variável é igual a “0”: Comando executado com sucesso!

```
1 # pws
2 # echo $?
3 # != 0
```

Quando o resultado é diferente de “0”, quer dizer que existiu algum problema na execução do comando.

Cada programa tem sua tabela e exceções, mas sempre retornam “0” quando o programa é bem executado. O comando “test”

O teste de condicionais (strings, matemáticas e em arquivos) em “Shell Script” é feito através do comando “test”. Vamos conferir algumas formas de testar condicionais.

Testando “strings”

```
1 # test "uva" = "uva"
2 # echo $?
3 # 0
```

```
1 # test "uva" = "banana"
2 # echo $?
3 # 1
```

Testando expressões matemáticas

```
1 # test 5 -eq 2
2 # echo $?
3 # 1
```

```
1 # test 2 -eq 2
2 # echo $?
3 # 0
```

Testando expressões em arquivos

```
1 # test -z $vazia
2 # echo $?
3 # 0
```

```
# var=valor
# test -z $var
# echo $?
# 1
```

Acima mostramos algumas formas de se testar as condicionais utilizadas dentro da estrutura “se”. Lembre-se que podemos usar as condicionais tanto dentro, quanto fora da estrutura “se”, depende do caso e do meio.

Abaixo podemos ver uma lista de operadores para nossa diversão.

Operadores de strings

Operadores	*Funções*
==	Igual
!=	Diferente

Operadores de matemáticos

Operadores	*Funções*
+	Soma
-	Subtração
*	Multiplicação
/	Divisão
>	Maior
>=	Maior ou Igual
<	Menor
<=	Menor ou Igual

Operadores para arquivos

Operadores	*Funções*
-e	Arquivo existe (exists)
-nt	Arquivo é mais novo que (newer than)
-ot	Arquio é mais antigo que (older than)
-d	É um diretório (directory)

Existem muitos outros operadores para que possamos dominar o mundo e consequentemente o sistemas UNIX com “shell script”. Estes são apenas os essenciais!

4.4 Utilizando a estrutura “if”

Abaixo alguns exemplos realmente práticos de como utilizar a estrutura “if”.

```
#!/bin/bash
## Primeiro script - Verificando se um usuário existe
echo "Digite usuário para consulta:"
read USER
REPLY=$(getent passwd | grep $USER)
if [ -z $REPLY ] ; then
   echo "Usuário $USER não existe!"
else
   echo "Usuário Existe"
fi
```

Ao invés de verificar se a variável que recebia o resultado do comando estava vazia, poderíamos ter utilizado o comando “test” antes da estrutura e checar apenas seu código de erro:

```
#!/bin/bash
## Primeiro script - Verificando se um usuário existe
#
echo "Digite usuário para consulta:"
read USER
REPLY=$(getent passwd | grep $USER)

test -z $REPLY
if [ $? -eq 0 ] ; then
  echo "Usuário $USER não existe!"
else
  echo "Pagamento em dia"
fi
```

4.5 Exemplos de script com IF

Exemplo 1: Verifica se um determinado usuário esta logado no sistema.

```
1  #!/bin/bash
2  clear
3  echo "Digite o nome do usuário"
4  read USER
5  if  who  | grep $USER > /dev/null
6  then
7  clear
8  echo $USER esta logado
9  else
10 clear
11 echo $USER não está logado
12 fi
```

Exemplo 2: Pesquisa uma palavra dentro de um arquivo.

```
1  #!/bin/bash
2  clear
3  echo "Escreva o nome do arquivo e a palavra a ser pesquisada:"
4  read file word
5  if grep $word $file > /dev/null
6  then
7  clear
8  echo "A palavra $word existe no arquivo $file."
9  fi
```

4451

Linux System Administration

www.4linux.com.br

Conteúdo

Automatizando Tarefas com Shell Script II

5.1 Introdução Teórica

Tarefas administrativas são, muitas vezes, longas e repetitivas. Podemos automatizar esses procedimentos através de "scripts". Como exemplo podemos utilizar o "case"para comandos de fluxo, tal como é o if, mas enquanto if testa expressões não exatas, o "case"vai agir de acordo com resultados exatos.

Temos o "while"para testa continuamente uma expressão, até que ela se torne falsa, e ainda contamos com o laço "for"que vai substituindo uma variável por um valor,e vai executando os comandos que são pedidos.

5.2 Utilizando a estrutura "case"

Outra estrutura bastante útil quando vários "if" precisam ser declarados é a estrutura "case".

case <valor> in <padrão1>) comandos ;; <padrão2>) comandos ;; <padrão3>) comandos ;; *) comandos ;; esac

Exemplo 1: Executar comandos dependendo do usuário digitado.

```
#!/bin/bash

clear
echo "Digite um nome de usuário"
read Usuario
case $Usuario in
aluno )
clear ; ls /etc ; cal ; date
;;
root )
clear ; whoami
;;
*)
clear
echo $Usuario não existe
;;
esac
```

Exemplo 2: Script que exibe informações do sistema.

```
#!/bin/bash
clear
echo "Escolha uma opção para informações da maquina (Digite o numero
   )"
echo "1-Horario do sistema"
echo "2-Tempo que o servidor esta ligado"
echo "3-Quantidade de usuário logados"
echo "4-Sair"
read ESC
case $ESC in
  1)
   H=$(uptime | awk -F" " '{ print $1 }')
   echo "Agora são $H"
```

```
13        ;;
14      2)
15       T=$(uptime | awk -F" " '{ print $3 }')
16   echo "O sistema esta $T ligado"
17        ;;
18      3)
19       U=$(uptime | awk -F" " '{ print $4 }')
20        echo "Existem $U atualmete logados"
21        ;;
22      4)
23        echo "Bye ..."
24        ;;
25  *)
26        echo "Opção invalida"
27         ;;
28  esac
```

5.3 Utilizando a estrutura “while”

Quando repetições são necessárias podemos utilizar estruturas de “looping” como “while” e “for”.

```
1  while [<expressão> ]; do
2   comandos
3  done
```

Exemplo 1: Verificar se um site esta no ar.

```
1  #!/bin/bash
2  clear
3  echo "Digite o endereço de um site"
```

```
4 read SITE
5 while ping -c1 $SITE > /dev/null 2>&1
6 do
7 echo "O site $SITE está no ar."
8 done
```

Exemplo 2: Cria quantos arquivos você indicar com uma determinada extensão.

```
1  #!/bin/sh
2  clear
3  echo "Digite o nome do arquivo"
4  read ARQ
5  clear
6  echo "Digite a extensão do arquivo"
7  read EXT
8  clear
9  echo "Digite o numero de arquivos criados"
10 read NUM
11 i=1
12 while [ $i -le $NUM ]
13 do
14    touch  $ARQ$i.$EXT
15    i=`expr $i + 1`
16 done
```

5.4 Utilizando a estrutura “for”

O “for” pode ser utilizado efetuar um “looping” no estilo do “while” ou para processar uma lista.

```
1 for VARIAVEL in <lista> ; do
```

```
2  comandos com a VARIAVEL
3 done
```

Exemplo 1: Compactar todos os arquivos do diretório atual.

```
1 #!/bin/bash
2 for i in ‘ls -1‘
3   do
4  tar -cvzf $i.tar.gz $i
5   done
```

Exemplo 2: Apaga todos os arquivos de uma determinada extensão.

```
1 #!/bin/bash
2 clear
3 echo "Digite a extensão dos arquivos que você quer apagar"
4 read ARQ
5 for i in *.$ARQ ; do
6 rm $i
7 done
```

Exemplo de Loop

Vamos criar um script que fará uma verificação de quais máquinas estão ativas na rede, para isso usaremos o comando ping.

```
1 #  ping 192.168.200.254
```

Usaremos algumas opções do comando ping para que ele não entre num loop, como acontece por padrão, e espere nossa interação para interrompe-lo:

```
1 # ping -c 2 -w 2 192.168.200.1
```

Iremos agora elaborar o shell script hostup.sh

```
1 #!/bin/bash
2  for IP in $(seq 1 15); do
3    ping -c 2 -w 2 192.168.200.$IP > /dev/null &&
4      echo "192.168.200.$IP - UP" ||
5      echo "192.168.200.$IP - DOWN"
6  done
```

Dê permissão de execução ao “script” e execute-o para testá-lo:

```
1 # cd /sbin
2 # chmod u+x hostup.sh
3 # hostup.sh
```

Programar em “shell script” é uma arte, e como na arte, em “shell” o limite é a sua imaginação. Para se aprofundar nesse assunto:

 http://jneves.wordpress.com/ http://aurelio.net

5.5 Transformar Shell Script em binário

Obtenha o arquivo compactado no site usando o comando wget:

```
1 # wget -c http://www.datsi.fi.upm.es/~frosal/sources/shc-3.8.6.tgz
```

Descompacte o arquivo:

```
1 # tar -xvzf shc-3.8.6.tgz
```

Copie o binário o shc para /usr/local/bin

```
1 # cp shc-3.8.6/shc /usr/local/bin/
```

Compile seu shell script usando o comando:

```
1 # shc -v -r -f script
```

Opções de linha de comando:

- **-v:** Modo verbose (mostra o que esta fazendo);
- **-r:** Gera um binário compatível com mais de um sistema;
- **-f:** Opção para o compilador buscar o arquivo;

Copie o arquivo binário para /bin (assim todos os usuários terão acesso)

```
1 # cp script.x /bin/script
```

Acerte as permissões do arquivo para que todos os usuários tenham acesso.

```
1 # chmod 755 /bin/script
```

Acesse o sistema com um usuario comum e digite o nome do script

4451

Linux System Administration

www.4linux.com.br

Conteúdo

Gerenciamento de dados SQL

6.1 Introdução ao SGBD

Antes de iniciar a criação de bancos de dados e tabelas, é preciso ter instalado na maquina um SGBD. Que consiste é um sistema gestor de base de dados, onde disponibiliza uma interface para que clientes (usuários), possam interagir com o banco de dados, de varias maneiras como inserir dados, pesquisar, excluir, entre outras tarefas.

Os comandos são executados usando a linguagem SQL (Structured Query Language - Linguagem de Consulta Estruturada), que é uma linguagem para banco de dados relacional, facilitando a interação com vários SGBDs, como por exemplo o Firebird, PostgreSQL, MySQL, entre outros.

 As provas do LPI irão cobrar comandos básicos de SQL.

6.2 Instalação do MySQL

1) Para os nossos testes, vamos utilizar o "MySql":

```
1  # aptitude install mysql-server
```

2) Ganhando acesso ao Banco de dados:

```
1 # mysql -u root -p
```

3) Visualizando “databases”:

```
1 mysql> show databases;
```

6.3 Criação do Banco de Dados e Tabelas

1) Criando “database”:

```
1 mysql> create database lpi;
```

2) Utilizando “database”:

```
1 mysql> use lpi;
```

3) Veremos agora como criar as tabelas, onde os dados serão armazenados. A síntese de criação de tabelas do MySQL é a seguinte:

CREATE [TEMPORARY] TABLE [IF NOT EXISTS] nome_tabela [(definição_create,...)] [table_options] [select_statement]

Novamente podemos nos valer do parâmetro opcional “IF NOT EXISTS” para executarmos o comando sem termos certeza da existência ou não da tabela. Para criarmos uma tabela executamos a seguinte parte da sintaxe: “CREATE TABLE nome_tabela”.

Mas é mais comum criarmos a tabela já acompanhada de seus campos (fields). Vamos criar no nosso exemplo a tabela LPI com os seguintes campos:

“id” - campo de identificação;

“certificacao” - título do nível da prova; “prova” - tipo de prova; “nota” - notas da prova;

A sintaxe completa do comando será:

```
mysql> create table lpi (
  -> id int(10) unsigned not null auto_increment,
  -> certificacao varchar(80) not null,
  -> prova int(4) unsigned not null,
  -> nota varchar(80) not null,
  -> primary key (id));
```

Os campos são definidos da seguinte forma:

nome_campo tipo [NULL | NOT NULL] [DEFAULT valor_padrão] [AUTO_INCREMENT]

No campo id por exemplo o tipo é “int(10)” com o modificador “unsigned”, ele não aceita valores nulos (not null) e é “auto_increment”, ou seja, seu valor é definido automaticamente, aumentando de 1 (um) em 1 (um) toda vez que um novo registro é adicionado. Para fazer uso desta funcionalidade é necessário adicionar o valor “0” ou “null” neste campo.

No campo “certificacao” escolhemos o tipo “varchar(80)” o que significa que este campo aceita caracteres alfanuméricos com no máximo 80 deles. O campo também não pode ser nulo.

4) Visualizando as tabelas:

```
1 mysql> show tables;
```

5) Liste a estrutura da tabela:

```
1 mysql> desc lpi;
```

Com o retorno do comando “desc” podemos ver quais os campos da tabela, qual o tipo dos campos, se aceitam ou não valores nulos, se existe uma chave primária, e se algum campo possui a propriedade “auto_increment”.

6.4 Inserir e pesquisar dados em tabelas

1) Para preencher os campos da tabela

```
1 mysql> INSERT INTO lpi (id, certificacao, prova, nota) VALUES ('01',
      'LPI', '101', '630');
```

2) Selecionar todos os registros da tabela “lpi” onde o campo “nota” possui o valor igual a 101:

```
1 mysql> SELECT * FROM lpi WHERE nota='101';
```

6.5 Atualizar campos e registros em tabelas

1) Adicione um novo campo na tabela:

```
1 mysql> alter table lpi
2     -> add nome varchar(80) not null;
```

2) Liste a estrutura da tabela:

```
1 mysql> desc lpi;
```

3) Selecionar todos os registros da tabela “lpi”:

```
1 mysql> select * from lpi;
```

4) Atualize o registro 1 onde o campo id possui o valor igual a 1:

```
1 mysql> update lpi set nome='fulano' where id='1';
```

6.6 Alterando privilégios

1) Adicionando usuário:

```
1 mysql> GRANT ALL PRIVILEGES ON lpi.* TO curso@localhost
2  IDENTIFIED BY 'cursolinux' WITH GRANT OPTION;
3  FLUSH PRIVILEGES;
```

Atribui todos os privilégios à todas as tabelas do banco “lpi” ao usuário “curso”, a partir da máquina “localhost”, cuja senha é “cursolinux”. O comando “FLUSH PRIVILEGES” atualiza as novas alterações no “daemon” do MySQL. Caso o usuário “curso” não exista, um novo usuário será criado.

2) Saindo do Mysql:

```
mysql> quit
```

3) Faça testes de login com o usuário “root” e o usuário “curso”:

```
# mysql -u root -p
# mysql -u curso -p
```

4451

Linux System Administration

www.4linux.com.br

Conteúdo

Administração de Usuários I

7.1 Registro de usuários no sistema

Há quatro arquivos básicos que dizem respeito à administração de usuários, sendo eles:

- **passwd ->** contém as informações dos usuários;
- **shadow ->** contém as informações das senhas dos usuários;
- **group ->** contém as informações dos grupos e usuários que fazem parte deles;
- **gshadow ->** contém informações a respeito das senhas de grupo.

Leitura Sugerida, para administração de usuários:

passwd -> man 5 passwd
shadow -> man 5 shadow

Leitura Sugerida, para administração dos grupos:

group -> man 5 group
gshadow -> man 5 gshadow

7.1.1 Arquivo /etc/passwd

Cada usuário cadastrado no sistema é identificado por uma linha no arquivo "/etc/-passwd". Os campos são separados pelo caractere ":" (dois pontos). O formato do arquivo "/etc/passwd" é o seguinte: usuario:x:1000:1000:User da Silva,8111-1234:/home/usuário:/bin/
Onde:

Campo 1 -> Login do usuário;

Campo 2 -> Referência da senha do usuário, pois ela fica armazenada em outro arquivo.

Campo 3 -> O "UID - User Identify" é o número de identificação do usuário. Essa identificação é dividida conforme a categoria dos usuários:

UID 0 -> É o número do usuário administrador "root".

Debian:

UID de 1 a 999 -> São os números para usuários de sistema.

UID de 1000 a 65535 -> São os números para usuários normais.

CentOS:

UID de 1 a 499 -> São os números para usuários de sistema.

UID de 500 a 65535 -> São os números para usuários normais.

Essas definições de usuários de sistema e usuários normais podem variar nas distribuições, somente o “UID 0” é padrão em todas as distribuições.

Campo 4 -> O “GID - Group Identity” é o número de identificação do grupo primário do usuário. Essa identificação é também dividida em 3 categorias como o UID:

GID 0 -> É o número do grupo administrador “root”.

Debian:

GID de 1 a 999 -> São os números para grupos de sistema.

GID de 1000 a 65535 -> São os números para grupos normais.

CentOS:

UID de 1 a 499 -> São os números para grupos de sistema.

UID de 500 a 65535 -> São os números para grupos normais.

Campo 5 -> Comentários e informações adicionais sobre o usuário;

Campo 6 -> Diretório pessoal;

Campo 7 -> Shell do usuário;

Usar o comando “getent”, é a maneira certa de se acessar arquivos de controle no GNU/Linux.

7.1.2 Arquivo /etc/shadow

As senhas dos usuários ficam armazenadas no arquivo “/etc/shadow” conhecido como “senhas sombras” (shadow passwords). As senhas ficam nele pois é um arquivo mais seguro que o arquivo “/etc/passwd”. No arquivo “/etc/passwd” qualquer usuário poderia visualizá-las e copiá-las para outro diretório ou máquina remota. Já o arquivo “/etc/shadow” tem suas permissões muito mais restritas, não permitindo que ele seja copiado e nem visualizado diretamente por um usuário comum. Isso é uma grande ajuda na questão de segurança, pois se as senhas estivessem no próprio ”/etc/passwd” seria muito fácil para um invasor com usuário comum, copiar esse arquivo para outro servidor e aplicar uma ferramenta de “brute force” para quebrar as senhas.

O suporte a senhas “shadow” costuma vir ativado por padrão em todas as distribuições. Em algumas delas, se forem instaladas no modo “expert”, é possível optar por ativar ou não esse suporte. É sempre recomendado deixar as senhas “shadow” ativadas.

Caso encontremos algum servidor GNU/Linux sem as senhas “shadow” configuradas, podemos utilizar o comando “pwconv” para ativá-las e “pwunconv” para desativá-las.

Em relação às senhas “shadow” e os comandos “pwconv” e “pwunconv”, muitas perguntas podem ser feitas na prova. Fique atento!

O arquivo “shadow” não trata somente a questão de segurança de senhas. Ele também trata de políticas de contas do usuário, como, por exemplo, por quantos dias a conta de um usuário é válida? Quando vai expirar? Quando deve ser a troca de senha? E alguns outros parâmetros que podem ser alterados manualmente ou usando o comando **“chage”**.

7.2 Levantamento de informações dos usuários

7.2.1 Comando chage

O comando "**chage**" configura algumas características da senha, como: data de validade, data de aviso de troca, dentre outras. O Nome do usuário no exemplo é "aluno", fique atento, porque esse comando é muito útil em seu dia-a-dia:

```
1 # chage -E 03/08/2012 aluno
2 # chage -l aluno
3 Última mudança de senha           : Set 12, 2011
4 Esta senha expira em            : nunca
5 Senha inativa                       : nunca
6 Conta expira em                 : Mar 08, 2012
7 Número mínimo de dias entre mudanças de senha : 0
8 Número máximo de dias entre mundanças de senha  : 99999
9 Número de dias de aviso antes da senha expirar  : 7
```

7.2.2 Comando id

O comando "id" mostra as informações de "UID", "GID" e grupos secundários dos usuários. Para ver essas informações do usuário corrente, fazemos da seguinte forma:

```
1 # id
```

Para ver as informações do usuário aluno utilize a seguinte sintaxe:

```
1 # id aluno
```

7.2.3 Comando groups

A lista dos grupos existentes no sistema é armazenada em /etc/group.

O uso do comando groups, sem parâmetros, faz com que o sistema informe os grupos dos quais o usuário é membro.

```
1 # groups
```

Para ver qual grupo o usuário aluno pertence utilize a seguinte sintaxe:

```
1 # groups aluno
```

7.2.4 Comando finger

O comando **“finger”** é mais amigável e nos traz maiores informações como: Login, Nome, Diretório “home”, “Shell”, número de e-mails não lidos e os horários das últimas autenticações que esse usuário realizou.

```
1 # finger aluno
```

7.2.5 Comando users

O comando **“users”** mostra de maneira bem simples os usuários que estão logados no sistema. A sintaxe do comando **“users”** é a seguinte:

```
1 # users
```

7.2.6 Comando who

O comando **“who”** mostra quais usuários estão logados na máquina. Traz informações adicionais sobre qual terminal está sendo utilizado, o momento e a partir de qual máquina foi feito o “login” de cada usuário.

```
1 # who
```

7.2.7 Comando w

O comando **“w”** é similar ao **“who”**, mas traz também informações sobre o que cada usuário está fazendo, tanto local quanto remotamente. Esse comando é muito útil para ver se não existem conexões indevidas em nosso sistema.

A sintaxe do comando **“w”** para visualizar todos os usuários logados é a seguinte:

```
1 # w
```

Para visualizar se o usuário aluno está logado a sintaxe é a seguinte:

```
1 # w aluno
```

7.3 Criando grupo

Um usuário sempre deve pertencer a um grupo primário, mas pode ser adicionado a grupos secundários, normalmente usado dentro de uma estrutura empresarial onde os usuários precisam pertencer a vários grupos para terem acessos a arquivos de outros setores.

7.3.1 Comando addgroup

Adiciona um grupo ao sistema. ou adiciona um usuário a um grupo.

Só funciona no Debian:

Adicione o grupo rede e grupo internet:

```
1 # addgroup rede
2 # addgroup internet
```

CentOS:

Não existe o comando.

7.3.2 Comando groupadd

O comando groupadd cria um novo grupo usando valores especificados na linha de comando e os valores padrões do sistema. O novo grupo será criado nos arquivos do sistema, conforme o requerido. Adicione o grupo aula:

```
1 # groupadd aula
```

7.3.3 Comando adduser

Só funciona no Debian:

Adicione o grupo empresa e social:

```
1 # adduser --group empresa
2 # adduser --group social
```

CentOS:

Não funciona no CentOS devido ao comando ser um link para o comando "use-radd".

7.4 Criando Usuários

Antes de criarmos um usuário, vamos definir o que conterá por padrão em seu diretório home, isto é definido no arquivo "/etc/skel", tudo o que estiver dentro deste diretório será adicionado ao home do usuário por padrão. Crie um diretório e um arquivo dentro do diretório /etc/skel:

```
1 # mkdir /etc/skel/importante
2 # touch /etc/skel/atividades.txt
```

7.4.1 Comando adduser

O comando "adduser"é um "script"customizado que trabalha como o comando "useradd". O "adduser"é bastante utilizado por administradores que precisam cadastrar usuários no formato tradicional, ou seja, com nome, senha e grupo, definindo, além disso ele também pode criar grupos e adicionar usuários em grupos.

No caso do CentOS, o comando “adduser” é um link para o comando “useradd”.

Este comando pode ser usado de várias formas, mas a sintaxe mais utilizada é a seguinte:

Debian:

```
1 # adduser [usuário]
```

Adicione o usuário mandark e a usuária meemee:

```
1 # adduser mandark
2 # adduser meemee
```

Dessa maneira ele adicionará o usuário, já pedindo para definir sua senha e as informações adicionais. Automaticamente, ele já cria um grupo com o mesmo nome do usuário e copia todos os arquivos que estão no diretório “/etc/skel” para o diretório “home” do usuário.

Visualize os grupos que o usuário mandark pertence e também os arquivos/diretórios criados a partir do /etc/skel em seu diretório home:

```
1 # id mandark
2 # ls /home/mandark
```

CentOS:

O comando adduser no CentOS é um link para o comando useradd. Veja o comando "useradd".

7.4.2 Comando useradd

Podemos também adicionar usuários através do comando “useradd”, que é um pouco mais complexo e precisa de alguns parâmetros a mais.

Adicione o usuário levinsky e o usuário leelee:

```
1 # useradd leelee
2 # useradd levinsky
```

Tente se logar com o usuário “levinsky” em um terminal e repare que não é possível, pois ainda não foi definida uma senha para ele. Adicione uma senha para o usuário:

```
1 # passwd levinsky
```

Agora tente se logar no terminal com o usuário levinsky e veja que, é possível após ser definida a senha. Mas tente se logar na parte gráfica e veja o que acontece, não é possível, pois o usuário não tem diretório home.

Para criar o usuário “deedee” com os principais atributos faça:

```
1 # useradd -m -s /bin/bash -u 3000 -g 100 -p ‘perl -e ’print crypt
    (123456, ”salt”)’‘    deedee
```

Acima criamos o usuário deedee, onde:

-m -> cria diretório home, caso ele não exista

-s -> shell do usuário

-u -> UID

-g -> GID

-p -> senha criptografada

perl -e ’print crypt(123456, “salt”)’ -> criptografar senha 123456 no formato crypt

Verifique o UID e GID do usuário vendas:

```
1 # id deedee
```

Logue-se com o usuário deedee no terminal e na parte gráfica.

7.5 Adicionando usuário ao grupo

7.5.1 Comando addgroup

O comando addgroup pode ser utilizado para adicionar um usuário a um grupo. Adicione o usuário mandark ao grupo rede, aula e aluno:

```
# addgroup mandark rede
# addgroup mandark aula
# addgroup mandark aluno
```

CentOS:

Não existe o comando.

7.5.2 Comando adduser

Só funciona no Debian:

O comando adduser também é utilizado para adicionar um usuário à um grupo, sua sintaxe é:

```
# adduser [usuário] [grupo]
```

Adicione o usuário mandark ao grupo empresa:

```
# adduser mandark empresa
```

Visualize os grupos que o usuário pertence:

```
# id mandark
```

CentOS:

O comando adduser no CentOS é um link para o comando useradd. Veja o comando "useradd".

7.5.3 Comando gpasswd

O comando **“gpasswd”** pode ser utilizado para definir a senha de um grupo. Utilizando a opção “-a” podemos adicionar um usuário a um grupo secundário.

Para adicionar um usuário a um grupo secundário a sintaxe é a seguinte:

```
1 # gpasswd -a [usuário] [grupo]
```

Adicione o usuário mandark ao grupo internet e social:

```
1 # gpasswd -a mandark internet
2 # gpasswd -a mandark social
```

Visualize:

```
1 # id mandark
```

7.6 Deletando usuário de um grupo

7.6.1 Comando deluser

Só funciona no Debian: O comando deluser também é utilizado para remover um usuário de um grupo: Delete o usuário mandark do grupo rede:

```
1 # deluser mandark rede
```

Visualize:

```
1 # id mandark
```

CentOS:

Não tem o comando.

7.6.2 Comando gpasswd

O comando gpasswd pode ser utilizado para remover um usuário de um grupo secundário. Para remover um usuário de um grupo secundário a sintaxe é a seguinte:

```
1 # gpasswd -d [usuário] [grupo]
```

Removendo o usuário mandark do grupo internet:

```
1 # gpasswd -d mandark internet
```

Visualize:

```
1 # id mandark
```

7.6.3 Comando delgroup

Remove um usuário de um grupo. **Só funciona Debian:**

Remova o usuário mandark do grupo social:

```
1 # delgroup mandark social
```

Visualize:

```
1 # id mandark
```

7.7 Removendo usuários

A remoção de usuários pode ser feita de duas formas. A primeira é mantendo o diretório “home” do usuário e a segunda, removendo também o “home”. É aconselhável que se remova o diretório do usuário para que um próximo usuário adicionado ao sistema não acabe como dono daquele diretório e tendo acesso a informações às quais ele não deveria ter. Isso pode acontecer porque a delegação de “UID’s” é sequencial. Mas para remover o usuário com o seu diretório, também é aconselhável, antes, fazer um backup de tudo o que aquele usuário possuía ou transferir todos os arquivos para o responsável. O usuário que será removido não pode estar logado.

7.7.1 Comando userdel

A sintaxe para remover o usuário e manter o seu diretório home é a seguinte:

```
1 # userdel [usuário]
```

Remova a usuária meemee:

```
1 # userdel meemee
```

Repare que a usuária meemee foi removida, mas seu diretório home não:

```
1 # ls -l /home
```

O problema aqui é que o próximo usuário que for criado, herdará o diretório pra si, veja:

```
1 # adduser herdeiro
2 # ls -l /home
```

Além de ser criado um diretório home, ele também herda o do usuário anterior, isto acontece porque os usuários são criados conforme os UID's disponíveis na sequência. Para remover o usuário e o seu diretório "home", é necessário utilizar a opção "-r" da seguinte forma:

```
1 # userdel -r [usuário]
```

Remova o usuário deedee e seu diretório home:

```
1 # userdel -r deedee
```

Verifique que foi deletado o diretório home do usuário deedee:

```
1 # ls -l /home
```

7.7.2 Comando deluser

Só funciona no Debian: O comando deluser deleta um usuário. Delete o usuário levinsky:

```
1 # deluser levinsky
```

Verifique que o diretório home do usuário não foi removido:

```
1 # ls -l /home
```

Adicione novamente o usuário levinsky:

```
1 # adduser levinsky
```

Delete o usuário levinsky e seu diretório home:

```
1 # deluser levinsky --remove-home
```

Verifique que o diretório home do usuário foi removido:

```
1 # ls -l /home
```

Adicione novamente o usuário levinsky:

```
1 # adduser levinsky
```

Delete o usuário levinsky e faça um backup do seu diretório home:

```
1 # deluser levinsky --remove-home --backup
```

Verifique que seu diretório home foi compactado:

```
1 # ls -l /home
```

Para complementar a seção removendo usuários com o comando deluser é muito interessante olhar o arquivo **"/etc/deluser.conf"**.

CentOS:

Não tem o comando.

7.8 Removendo grupos

7.8.1 Comando deluser

O comando deluser pode também deletar um grupo, desde que este não seja o grupo primário de um usuário. Vamos deletar o grupo criado anteriormente chamado empresa:

```
1 # deluser --group empresa
```

7.8.2 Comando groupdel

Apaga um grupo do sistema. Quando é usado, este comando apaga todos os dados do grupo especificado dos arquivos de contas do sistema. Não é possível remover o grupo primário de um usuário. Remova o usuário primeiro.

Visualize que o usuário mandark pertence ao grupo **aula** que criamos anteriormente:

```
1 # id mandark
```

Removendo o grupo aula:

```
1  # groupdel aula
```

Visualize que o usuário mandark não pertence mais ao grupo aula:

```
1 # id mandark
```

7.8.3 Comando delgroup

Remove um grupo do sistema. Remova o grupo internet:

```
1 # delgroup internet
```

CentOS:

Não existe o comando.

4451

Linux System Administration

www.4linux.com.br

Conteúdo

Administração de Usuários II

8.1 Modificando Usuários

A modificação de usuários é limitada ao usuário “root”. Iremos aprender aqui como mudamos alguns parâmetros que são necessários no dia-a-dia, como troca de senhas, grupos e controle de “login”.

8.1.1 Comando passwd

Depois do usuário ter sido criado podemos usar alguns comandos para modificar sua conta. O primeiro será o “passwd” que possibilita adicionar ou modificar a senha de um usuário. As principais sintaxes que podem ser utilizadas nesse comando estão descritas abaixo.

Para modificar a senha do usuário corrente:

```
1 # passwd
```

Caso esteja modificando a senha de um usuário normal, primeiro será solicitada a senha corrente para permitir a definição de uma nova senha. Isso não

acontece com o usuário “root”, que pode definir a nova senha diretamente, tanto para ele quanto para os outros usuários.

Para modificar a senha do usuário mandark:

```
1 # passwd mandark
```

Para bloquear a conta do usuário mandark:

```
1 # passwd -l mandark
```

Tente se logar com o usuário mandark, não é possível, pois sua conta está bloqueada. Para desbloquear a conta do usuário mandark:

```
1 # passwd -u mandark
```

Agora o usuário já pode se logar.

8.1.2 Comando usermod

Para modificar nomes de grupos do sistema, utilizamos o comando:

```
1 # groupmod -n [novo-nome]  [nome-grupo]
```

```
1 groupmod -n vendas rede
```

Outro comando que pode ser utilizado para modificar parâmetros do usuário é o **"usermod"**. Ele possibilita alterar qualquer tipo de informação relativa ao usuário. Um dos parâmetros que pode ser modificado é o grupo primário, usando-se a opção "-g". Com a opção "-G", podemos alterar os grupos secundários.

A sintaxe para modificar o grupo primário de um usuário é a seguinte:

```
1 # usermod -g [grupo] [usuário]
```

Verifique o grupo primário do usuário mandark:

```
1 # id mandark
```

Altere seu grupo primário para audio:

```
1 # usermod -g audio mandark
```

As alterações podem ser visualizadas no arquivo **"/etc/passwd"** no campo "GID", ou diretamente no arqu1ivo **"/etc/group"**. Para trocar todos os grupos secundários pelos grupos aluno e vendas, faça:

```
1 # usermod -G aluno,vendas mandark
```

Visualize:

```
1 # id mandark
```

Altere seu grupo primário para mandark:

```
1 # usermod -g mandark mandark
```

Para mudarmos o campo de informações dentro do arquivo “/etc/passwd”, precisamos usar o comando “usermod” com a opção “-c”.

```
1 # usermod -c "Dpto vendas" mandark
2 # getent passwd | grep mandark
3 mandark:x:1001:1001:Dpto mandark:/home/mandark:/bin/bash
```

Outras opções do comando:

-d diretório [-m] : cria um novo diretório home para o usuário. A opção -m faz com que o diretório atual do usuário seja movido para o novo diretório.

-e mm/dd/yy : altera a data de expiração da conta do usuário.

-l nome : altera o nome de identificação do usuário (o usuário não pode estar logado).

-s shell : altera o shell do usuário.

-u uid : altera o número de UID do usuário.

-L : bloqueia a conta acrescentando um “!” no início da linha do usuário no arquivo “/etc/passwd”

-U : desbloqueia a conta

8.2 Alteração do Dono e Grupo

Como já vimos, cada arquivo e diretório possui um dono e um grupo. Para alterá-los podemos utilizar os comandos “chown” e “chgrp” como nos exemplos a seguir.

Vamos criar o diretório home do usuário leelee que ainda não existe:

```
1 # mkdir /home/leelee
```

Como criamos como usuário root, o dono e o grupo do diretório criado pertencem ao usuário root e grupo root:

```
1 # ls -ld /home/leelee
2 drwxr-xr-x  2 root    root    4096 2011-11-08 18:00 leelee
```

Temos que mudar o usuário e grupo para o usuário leelee e seu grupo primário leelee. Alterando apenas o grupo de root para leelee:

```
1 # chgrp leelee /home/leelee
```

Visualize:

```
1 # ls -ld /home/leelee
2 drwxr-xr-x  2 root    leelee    4096 2011-11-08 18:00 leelee
```

Outra forma de trocar apenas o grupo é com o comando chown, veja sua sintaxe:

```
1 # chown [dono.grupo] [arquivo] -> troca dono e grupo
```

```
# chown [dono:grupo] [arquivo] -> troca dono e grupo
# chown [dono] [arquivo] -> troca apenas o dono
# chown [dono.] [arquivo] -> troca dono e grupo ”mesmo grupo primá
    rio do dono”
# chown [dono:] [arquivo] -> troca dono e grupo ”mesmo grupo primá
    rio do dono”
# chown [.grupo] [arquivo] -> troca o grupo
# chown [:grupo] [arquivo] -> troca o grupo
```

Troque o dono para leelee:

```
# chown leelee /home/leelee
```

Visualize:

```
# ls -ld /home/leelee
drwxr-xr-x  2 leelee    leelee    4096 2011-11-08 18:00 leelee
```

Troque a senha do usuário leelee:

```
# passwd leelee
```

Agora se logue na parte gráfica como usuário leelee. Para poder alterar o dono/-grupo de arquivos e diretórios dentro do diretório utilize o parâmetro **“-R”** para fazer a alteração recursiva

8.3 Introdução a tipos de permissões

O GNU/Linux é um **sistema multi-usuário** e portanto, possui um esquema de permissões que provê a privacidade e/ou compartilhamento de arquivos entre usuários. Na verdade, esse esquema de permissões é parte fundamental do sistema. Neste capítulo, iremos aprender sobre ele e também como criar e remover contas de usuários.

Quando começamos a trabalhar com usuários no sistema GNU/Linux podemos dividi-los em três categorias:

- **Usuário Administrador (Super Usuário):** usuário conhecido como “root” no sistema. É esse usuário que controla todo o sistema e não possui nenhuma restrição. Mas devemos ter uma certa cautela ao usá-lo pois com qualquer deslize podemos danificar todo o sistema;

- **Usuários de Sistema:** são aqueles que não precisam “logar” no sistema, são utilizados para controlar serviços. Esses usuários não devem possuir senhas nem “Shell” válida. Um exemplo desses usuários é o “www-data” que é usado exclusivamente para controlar o servidor web “Apache”;

- **Usuários comuns:** são utilizados para trabalhar no sistema GNU/Linux. São contas criadas para aqueles que utilizam ou operam o sistema. É sempre aconselhável que cada usuário comum ou administrador tenha sua própria conta e só utilize a conta “root” para administração do sistema.

Tanto para o usuário “root”, quanto para o usuário comum, é sempre aconselhável ter uma boa política de criação de senhas, para que um possível invasor não se aproveite de um usuário com uma senha fraca. Até mesmo um usuário comum, precisa tomar cuidado com a sua senha, pois esse seria o primeiro passo para o invasor escalar privilégios no sistema, e virar o usuário administrador “root”. Evite usar senhas com datas de aniversário, casamento e outras datas que são fáceis de serem descobertas. Evite usar palavras listadas em um dicionário. Uma boa

dica é mesclar a senhas com letras maiúsculas e minúsculas, números e caracteres especiais.

Alguns sistemas GNU/Linux podem ter usuários que chamamos de administradores. Esses usuários não vêm configurados por padrão, eles são usuários normais mas que possuem alguns privilégios a mais em algumas aplicações.

Para que os usuários comuns e o “root” tenham acesso ao sistema e consigam trabalhar normalmente, são necessários 5 elementos.

- Nome;
- Senha;
- Diretório Home;
- Shell;
- Grupo Primário;

Devemos ter em mente que um usuário sempre deve estar vinculado a um grupo, pois isso afeta diretamente a questão de permissões dentro do sistema.

8.4 Permissões

Cada arquivo no sistema possui três permissões básicas:

r (4) -> read - para leitura;

w (2) -> write - para escrita;

x (1) -> execute - para execução;

A cada permissão é atribuído um valor, mostrado entre parênteses, que será utilizado para a definição de permissões.

Além disso, cada arquivo contém três conjuntos de permissões, sendo elas:

permissão do dono (u) - “user” do arquivo;

do grupo (g) - “group” ao qual o arquivo pertence;

outros (o) - “others” aqueles que não pertencem ao grupo e não são os donos do arquivo;

Sendo assim, considere a seguinte saída do comando ls -l, para um arquivo: permissão do arquivo:

```
1 -rw-r--r-- 1 root root 0 Jan 15 09:52 arquivo
```

E para um diretório: permissão do diretório:

```
1 drwxr-xr-x 2 root root 4096 Jan 15 09:52 diretório
```

Vamos entender o que essas linhas significam. O primeiro caractere pode ser:

“-” -> indicando a listagem de um arquivo comum”;

d -> indicando um diretório;

l -> indicando um “link” simbólico;

p -> indicando um “pipe” nomeado;

s -> indicando um “socket”;

c -> indicando um dispositivo de caractere;

b -> indicando um dispositivo de bloco.

Os próximos três conjuntos de três caracteres indicam as permissões do usuário dono do arquivo, permissões de grupo e permissões para outros usuários. Nesses três conjuntos, se o caractere encontrado for um “-” (hífen) significa que a permissão está ausente, ou seja, não há a respectiva permissão. Se alguma ou todas as letras **(r, w e x)** forem encontradas, indicará as permissões que o arquivo tem permissões definidas.

Seguindo o conjunto de permissões, há um número que indica a quantidade de “links” simbólicos que o arquivo ou diretório tem. Após o número de “links”, vem a indicação do usuário dono do arquivo, seguido do grupo ao qual ele, o arquivo ou diretório, pertence.

Já criado aluno, mandark no grupo aluno

Vamos criar um arquivo para testes, se logue como usuário aluno:

```
1 $ cd /tmp
2 $ touch arquivo
```

Visualize a permissão do arquivo criado:

```
1 $ ls -l /tmp/arquivo
2 -rw-r--r-- 1 aluno aluno 0 2011-11-04 12:17 arquivo
```

Sua permissão para o dono é **Leitura e escrita: r w -**

Sua permissão para grupo é **Somente leitura: r - -**

Sua permissão para outros é **Somente leitura: r - -**

O comando para trocar as permissões é o **chmod.** Há duas sintaxes possíveis: literal e octal.

8.4.1 LITERAL

```
$ chmod u-rw /tmp/arquivo
```

O parâmetro **“u-rw”** é que define o esquema de permissões. A primeira letra indica para qual(is) usuário(s) as permissões estão sendo alteradas. Usamos a letra **“u”** para indicar o próprio dono, **“g”** para indicar o grupo, “o” para outros e ainda a letra “a” para indicar **todos.**

O caractere seguinte poderá ser um sinal de “=” para deixar a permissão igual à que se deseja, “+” para garantir a permissão ou “-” para retirar a permissão. Por fim, detalhamos a permissão: A letra **“r”** significa leitura, **“w”** escrita e **“x”** execução, como era de se esperar.

Assim, o exemplo anterior retira as permissões de leitura e escrita para o usuário dono do arquivo.

Verifique novamente a permissão do arquivo:

```
$ ls -l /tmp/arquivo
----r--r-- 1 aluno aluno 0 2011-11-04 12:17 arquivo
```

Sua permissão para o dono é **Nenhuma: - - -**

Sua permissão para grupo é **Somente leitura: r - -**

Sua permissão para outros é **Somente leitura: r - -**

As permissões seguem uma ordem dono, grupo, outros, ou seja, se você é o dono as permissões que se encaixam no seu perfil é a de dono, mesmo que você pertença ao grupo, as permissões de dono prevalecem.

Tente visualizar o arquivo como usuário aluno:

```
1 $ cat /tmp/arquivo
```

Agora se logue com o usuário mandark, que pertence ao grupo aluno, em outro terminal e tente ler o arquivo:

```
1 $ cat /tmp/arquivo
```

As permissões de grupo permitem que ele abra o arquivo para leitura, mas não para escrita:

```
1 $ echo oi >> /tmp/arquivo
```

Vejamos mais um exemplo, como usuário aluno faça:

```
1 $ chmod g+w /tmp/arquivo
```

Este comando adiciona a permissão de escrita para os usuários que fazem parte do mesmo grupo ao qual o arquivo pertence. As demais permissões não são alteradas.

Se logue com o usuário mandark que pertence ao grupo aluno e tente escrever no arquivo.

```
1 $ echo tchau >> /tmp/arquivo
2 $ cat /tmp/arquivo
```

Fazendo o teste com diretório, como usuário aluno crie um diretório:

```
1 $ mkdir /tmp/diretorio
```

Verifique a permissão do diretório:

```
1 $ ls -ld /tmp/diretorio
2 drwxr-xr-x 2 aluno aluno 4096 2011-11-04 17:02 diretorio/
```

Crie um arquivo dentro do diretório:

```
1 $ touch /tmp/diretorio/novo
```

Visualize e acesse o diretório:

```
1 $ ls -l /tmp/diretorio
2 $ cd /tmp/diretorio
```

Retire o acesso ao diretório para todos os usuários:

```
1 $ cd ..
2 $ chmod a-x /tmp/diretorio
```

Agora tente acessar o diretório:

```
$ cd /tmp/diretorio
```

Não é possível isso porque o x é o responsável por dar acessos ao diretório, mas ainda é possível visualizar o conteúdo do diretório:

```
$ ls /tmp/diretorio
```

Visualize as permissões do diretório e do arquivo:

```
$ ls -ld /tmp/diretorio
drw-r--r-- 2 aluno aluno 4096 2011-11-04 17:02 diretorio/
$ ls -l /tmp/diretorio/novo
-????????? ? ? ? ?                ? novo
```

8.4.2 OCTAL

A segunda sintaxe é a forma numérica. Neste caso, o parâmetro que define as permissões é composto de três números de 0 a 7, que correspondem às permissões para o usuário dono, para o grupo e para outros. Cada número é formado pela soma das permissões atribuídas, sendo que **execução vale 1**, **escrita vale 2 e leitura 4.**

r (4)	w (2)	x (1)	Total	Permissões
0	0	0	**0**	**- - -**
0	0	1	**1**	**- - x**
0	1	0	**2**	**- w -**
0	1	1	**3**	**- w x**
1	0	0	**4**	**r - -**
1	0	1	**5**	**r - x**
1	1	0	**6**	**r w -**
1	1	1	**7**	**r w x**

Vejamos um exemplo:

```
# chmod 640 /tmp/arquivo
```

Neste caso, estamos atribuindo a permissão de:

leitura e escrita 6 (r=4 + w=2) ao usuário dono

leitura 4 (r=4) ao grupo

0 (sem permissões) à outros usuários

É importante observar que quando usamos a forma literal, alteramos apenas o parâmetro especificado, não alterando as demais permissões. Já na forma numérica, alteramos todas as permissões simultaneamente.

8.4.3 Exemplos de permissões

Comando para atribuir permissão total a um arquivo chamado naofazer:

```
# touch /tmp/naofazer
# chmod 777 /tmp/naofazer
```

Ou

```
1 # chmod a+rwx /tmp/naofazer
```

Ou

```
1 # chmod u+rwx,g+rwx,o+rwx /tmp/naofazer
```

Verifique a permissão:

```
1 # ls -l /tmp/naofazer
2 -rwxrwxrwx 1 root root 0 2011-11-06 22:17 naofazer
```

Evite dar permissão total para um arquivo ou diretório a não ser que seja realmente preciso.

Não se deve fazer isso em nenhum tipo de arquivo, isso é apenas um exemplo!!!

Comando para retirar a permissão de escrita de todos os usuários do arquivo "naofazer":

```
1 # chmod 666 /tmp/naofazer
```

Ou

```
1 # chmod a-x /tmp/naofazer
```

Ou

```
# chmod u-x,g-x,o-x /tmp/naofazer
```

Visualize:

```
# ls -l /tmp/naofazer
```

Comando para alterar a permissão padrão do arquivo “arquivo” para que todos os usuários apenas possam lê-lo.

```
# chmod 444 /tmp/arquivo
```

Visualize:

```
# ls -l /tmp/arquivo
```

Utilizando o exemplo anterior do modo Literal, vamos alterar a permissão recursivamente, dando permissão total somente para o dono:

ANTES:

```
# ls -ld /tmp/diretorio
drw-r--r-- 2 aluno aluno 4096 2011-11-04 17:02 diretorio/

# ls -l /tmp/diretorio/novo
-rw-r--r-- 1 aluno aluno 0 2011-11-04 17:40 novo

# chmod -R 700 /tmp/diretorio
```

DEPOIS:

```
1 # ls -ld /tmp/diretorio
2 drwx------ 2 aluno aluno 4096 2011-11-04 17:02 diretorio/
3
4 # ls -l /tmp/diretorio/novo
5 -rwx------ 1 aluno aluno 0 2011-11-04 17:40 novo
```

8.5 Umask

O **“umask”** altera o valor da máscara de criação de arquivos e diretórios. Essa “máscara” é utilizada para definir o “permissionamento” padrão de um arquivo ou diretório quando ele é criado.

```
1 Debian:
2
3   O valor padrão da "umask" fica armazenada no arquivo "/etc/login.
      defs". Caso não exista adicione umask <valor\_da\_umask>.
4
5 Ex:
6
7 umask 0022
8
9   O primeiro "0" significa modo octal, pode-se passar o valor em
      hexadecimal colocando "0x" como prefixo.
10 CentOS:
11
12   O valor padrão da "umask" fica armazenada no arquivo "/etc/bashrc"
```

Para visualizar a umask atual:

```
1 # umask
```

No Debian: a umask padrão é de “0022” No CentOs: a umask varia de acordo com o usuário, quando seu UID é maior do que 99 e seu grupo tem o mesmo número do UID, sua umask é 0002, caso contrário será 0022.

8.5.1 Cálculo da umask

Para Diretório:

Para calcular a “umask” para um diretório, pegue a permissão total que um diretório pode chegar, “777”. Subtraia “a sua umask atual”.

EX: umask 022

777 permissão máxima para um diretório - 022 umask atual = 755 permissão do diretório a ser criado

Ex: umask 033

777 permissão máxima para um diretório - 033 umask atual = 744 permissão do diretório a ser criado Para Arquivo:

Para calcular a “umask” para um arquivo, saiba que um arquivo não pode ser criado com permissão de execução por padrão, esta permissão só pode ser passada para ele manualmente. Logo a permissão do arquivo não pode ser ímpar, porque o bit de execução vale 1. Como calcular? Ex: umask 022

777 Permissão máxima para um arquivo - 022 umask atual = 755 permissão para o

arquivo, mas lembre-se não pode ter permissão de execução, então subtraia 1 dos bits que sejam ímpares:

755 - 111 retirando os bits de execução = 644 permissão real do arquivo

Ex: umask 033

777 Permissão máxima para um arquivo - 033 umask atual = 744 permissão para o arquivo, mas lembre-se não pode ter permissão de execução, então subtraia 1 dos bits que sejam ímpares:

744 - 100 retirando os bits de execução = 644 permissão real do arquivo

Repare que a permissão não muda para o arquivo com a umask 022 e umask 033.

Lembre-se da regra de cálculo de "umask". Pensar da forma que o sistema funciona pode te confundir na prova: Para diretórios: Sempre substituir de 777; Para arquivos: Verificar o "umask". Se o número for ímpar, subtrair somente onde temos execução, em números pares mantemos os números.

8.6 Permissões Especiais

Há um conjunto especial de permissões, conhecido também como "bits" especiais, sendo eles:

Nome	Significado	Valor
SUID	Set User Id Bit	4
SGID	Set Group ID Bit	2
Sticky Bit	Sticky Bit	1

O **"SUID bit"** é atribuído a um arquivo binário com permissão de execução, quando

desejamos que um usuário qualquer execute o comando com as permissões do usuário dono do comando. Se esse comando pertencer ao usuário “root” um usuário qualquer irá executá-lo com as permissões de “root” desde que tenha permissões para executá-lo. Por esse motivo o “SUID” constitui uma grande ameaça de segurança e sua utilização deve ser bastante cautelosa.

O **“SGID bit”** é geralmente atribuível a diretórios. Quando um arquivo é criado dentro de um diretório com “SGID bit” ativado, o conteúdo gravado dentro do diretório irá herdar o grupo do diretório e não o grupo do usuário que criou tal conteúdo. Este “bit” especial é muito útil quando utilizamos diretórios para grupos de trabalhos e em servidores de arquivos.

O **“Sitcky bit”** era bastante utilizado para realizar otimizações de acesso a conteúdos, entretanto, a partir da série 2.6 do kernel do Linux essa tarefa é realizada diretamente pelo kernel. A única utilidade desse “bit”, atualmente, é fazer diretórios de utilização comum a todos os usuários, como no “/tmp”. Quando esse “bit” está ativo em um diretório, todo conteúdo criado dentro dele pertencerá ao criador do conteúdo e por mais que ele atribua a esse conteúdo permissões totais para todos os usuários, o único que poderá excluir o arquivo ou diretório será o próprio dono ou o “root” ou ainda o dono do diretório que tem a permissão. Para atribuirmos esses “bits” especiais, procedemos da mesma forma que nas permissões comuns, somando os valores e utilizando o comando “chmod”, mas agora utilizando quatro números, o primeiro número sendo o “bit” especial, seguido dos três da permissão padrão.

Veja o exemplo abaixo:

```
1 # chmod 4000 /tmp/a
2 # chmod 2000 /tmp/b
3 # chmod 1000 /tmp/c
4
5 # ls -l /tmp
6 ---S------ 1 caio caio 0 2008-07-21 13:50 a
7 ------S--- 1 caio caio 0 2008-07-21 13:50 b
8 ---------T 1 caio caio 0 2008-07-21 13:50 c
```

O "bit" especial para o campo de permissões do dono é o "SUID" representado por "s" ou "S". Para o grupo é "SGID" também representado por "s" ou "S". Já o campo de permissões de outros usuários, o "Sticky BIT", é representado por "t" ou "T".

Veja que quando o arquivo ou diretório não tem permissão de execução, o "bit" especial é representado por uma letra "S" (Upper Case), e quando possuem uma permissão de execução, o "bit" especial é apresentado como "s" (Lower Case). O mesmo acontece com o "Sticky bit", mas com a letra "t" e "T".

Exemplo dos bits especiais com permissão de execução:

```
1 # chmod 4100 /tmp/a
2 # chmod 2010 /tmp/b
3 # chmod 1001 /tmp/c
4
5 # ls -l
6 ---s------ 1 root root 0 2008-07-21 13:50 a
7 ------s--- 1 root root 0 2008-07-21 13:50 b
8 ---------t 1 root root caio 0 2008-07-21 13:50 c
```

Todas as permissões especias que não contiverem execução são maiúsculas. "S" e "T".

Exemplos:

Suid Bit:

Podemos usar como exemplo o comando "passwd".

```
1 # ls -l /usr/bin/passwd
2 -rwsr-xr-x 1 root root 31640 2008-06-12 20:39 /usr/bin/passwd
```

Os nossos usuários comuns só podem mudar sua senha pois o comando passwd está com o “bit SUID” ativado.

Desabilite o Suid:

```
1 # chmod 755 /usr/bin/passwd
2 # ls -l /usr/bin/passwd
```

Agora se logue como aluno e tente mudar sua senha:

```
1 $ passwd
```

Não é possível, pois o aluno não tem permissão de escrita nos arquivos:

```
1 /etc/passwd e /etc/shadow.
```

Como root, volte a permissão original:

```
1 # chmod 4755 /usr/bin/passwd
2 # ls -l /usr/bin/passwd
```

SGID Bit:

Crie um diretório com permissão total para qualquer usuário:

```
1 # mkdir /teste
2 # chmod 777 /teste
```

Agora qualquer usuário tem acesso ao diretório teste.

Como usuário aluno crie um arquivo no diretório teste:

```
$ touch /teste/numero1
```

Verifique quem é o dono e quem é o grupo do arquivo criado:

```
$ ls -l /teste
-rw-r--r-- 1 aluno aluno 0 2011-11-07 11:01 numero1
```

Verifique que o grupo é o mesmo do usuário.

Agora como root altere a permissão do diretório teste, adicionando o SGID Bit:

```
# chmod 2777 /teste
```

Novamente como usuário aluno, crie um novo arquivo dentro do diretório teste:

```
$ touch /teste/numero2
```

Verifique quem é o dono e quem é o grupo do arquivo criado:

```
$ ls  -l  /teste
-rw-r--r-- 1 aluno aluno   0 2011-11-07 11:01 numero1
-rw-r--r-- 1 aluno root    0 2011-11-07 11:19 numero2
```

Visualize que o grupo agora não é mais o do usuário e sim o mesmo do diretório.

Stick Bit:

Como root modifique novamente a permissão do diretório teste, adicione o Stick Bit:

```
1 # chmod 1777 /teste
```

Agora modifique as permissões dos arquivos dentro dele dando permissão total para todos os usuários:

```
1 # chmod  777 /teste/*
```

Agora se logue como usuário mandark, visualize as permissões do diretório e dos arquivos:

```
1 $ ls -ld /teste
2 drwxrwsrwt 2 root root 4096 2011-11-07 11:19 /teste
3 $ ls -l /teste
4 -rwxrwxrwx 1 aluno aluno 0 2011-11-07 11:01 /teste/numero1
5 -rwxrwxrwx 1 aluno root 0 2011-11-07 11:19 /teste/numero2
```

Agora que você viu que tem permissão total nos arquivos, tente deletar algum arquivo dentro do diretório teste:

```
1 $ rm /teste/numero1
```

Apesar da permissão total no arquivo, não é possível remover devido a permissão do diretório com Stick Bit.

4451

Linux System Administration

www.4linux.com.br

Conteúdo

Quotas de Disco

8.1 Introdução Teórica

A utilização de um sistema de quotas é um assunto tão importante quanto dividir o disco rígido em partições. O sistema de quotas serve para limitarmos a quantidade de blocos e “inodes” que um usuário ou grupo pode utilizar em uma determinada partição.

Imagine um HD com 100MB de “home” e 10 usuários. Se não utilizarmos um sistema de quota por número de blocos é possível que um dos usuários resolva fazer o download de um arquivo de 90MB utilizando 90% do espaço disponível, fazendo com que os outros usuários tenham que dividir os outros 10MB livres. Se aplicarmos um sistema de quotas, podemos definir que cada usuário utilizará no máximo 10MB, de forma que cada um terá o mesmo espaço disponível, tornando a divisão justa.

Em um cenário como este, resolvemos parte do problema, pois o usuário é capaz de criar um número, suficientemente grande de arquivos com tamanho zero de forma que ele não ocupe os 10MB atribuídos a ele mas estoure o número máximo de “inodes” que o sistema de arquivos dispõe, impossibilitando assim, que outro usuário grave qualquer coisa neste sistema de arquivos, mesmo que haja espaço livre.

O sistema de quotas é uma funcionalidade do “filesystem” e do kernel, sendo assim, ambos têm que serem capazes de suportá-lo. Uma vez que o “filesystem” suporta quotas, devemos adicionar os parâmetros de montagem, “usrquota” e “grpquota” ao “filesystem” que utilizaremos com esse sistema. Isso é feito no arquivo “/etc/fstab”. Além disso, temos que criar, na raiz desses “filesystems”, os arquivos de controle,

chamados “aquota.user” e “aquota.group”.

Uma vez criada essa estrutura, basta editar os arquivos de controle de quotas e distribuir as quantidades de forma apropriada. A quota somente pode ser aplicada por partições.

Instale o pacote de quota:

 # aptitude install quota

 # yum install quota

Edite o arquivo “/etc/fstab” e inclua as opções de quota por usuário e por grupo em “/home”:

```
1 UUID=12e9cf3f-99b3-4e8e-8079-d4337b2ce9c8  /home ext3  defaults,
     usrquota,grpquota 0 2
```

Remonte o “/home” para que as alterações sejam efetuadas:

```
1 # mount -o remount /home
```

Verifique se as opções de quota foram aplicadas:

```
1 # mount
```

Crie os arquivos de quota na raiz da partição que receberá o sistema de quotas:

```
1 # quotacheck -cug /home
```

-c -> cria arquivos de quota

-u -> checa quotas de usuários

-g -> checa quotas de grupos Caso dê erro:

-f -> força checagem das quotas

-m -> força checagem no filesystem montado como leitura e escrita , não remonta o filesystem como somente leitura em caso de erro.

Certifique-se de que os arquivos de controle de quota foram criados: "aquota.group"e "aquota.user".

```
1 # ls -l /home
```

Habilite a quota na partição /home:

```
1 # quotaon /home
```

Caso queira desabilitar a quota na partição o comando é:

```
1 # quotaoff /home
```

Verifique se o sistema de quota está ativo, listando as suas informações de quotas para usuários:

```
1 # repquota -va
```

Verifique se o sistema de quota está ativo, listando as suas informações de quotas para grupos:

```
1 # repquota -vag
```

8.1.1 Quotas por Usuário

Vamos definir qual a quantidade de recursos do HD que cada usuário poderá utilizar. Vamos impor que o usuário "mandark" poderá utilizar até 50MB com um limite máximo de 60MB ou 100 arquivos com o limite máximo de 110 arquivos.

Editando a quota do usuário mandark:

```
1 # edquota -u mandark
```

Dentro do "edquota", faremos as configurações para que a quota do usuário mandark, seja de 50MB e ele possa criar 100 arquivos, com um limite máximo acima da sua quota de 10MB e 10 arquivos. Altere o arquivo para que fique como mostrado a seguir:

```
1   Disk quotas for user mandark (uid 1001):
2  Filesystem blocks  soft  hard  inodes  soft  hard
3  /dev/sda3  0     50000   60000   0   100   110
4         ^-------NÃO MEXER-------^
```

Onde:

Filsesystem -> partição onde será aplicada a quota.

blocks -> tamanho real utilizado em KBytes. (não é possível alterar)

soft -> limite da quota de espaço disponível para gravação, ao ultrapassar este limite o usuário estoura sua quota.

hard -> limite máximo permitido de espaço disponível para gravação, após exceder sua quota, existe um período de tempo chamado "grace time" para uso deste limite.

inodes -> tamanho real utilizado em número de arquivos. (não é possível alterar)

soft -> limite da quota de número de arquivos que podem ser criados, ao ultrapassar este limite o usuário estoura sua quota.

hard -> limite máximo permitido de número de arquivos que podem ser criado, após exceder sua quota, existe um período de tempo chamado "grace time" para uso deste limite.

Verifique se o limite já está aplicado:

```
1 # repquota -v -a
```

Já que fizemos a gentileza de determinar que o usuário poderá usar 10MB ou 10 arquivos a mais caso ele estoure a sua quota, devemos determinar também por quanto tempo ele poderá usar esse espaço a mais.

Determine que os usuários terão 5 dias de "grace period":

```
1 # edquota -t
```

Caso o usuário estoure sua quota ele tem um tempo(grace period) antes de sua

conta ser bloqueada para apagar os arquivos necessários para utilizar o limite de sua quota.

Consulte a quota do usuário “mandark”.

```
1 # quota -u mandark
```

Efetue login em outro terminal utilizando o usuário “mandark” e vamos rodar um comando para encher o disco:

```
1 $ yes > a
```

Depois que a quota estourou, volte ao terminal do “root”, examine o status da quota e veja se a quota do usuário mandark está estourada por espaço utilizado:

```
1 # repquota -va
```

Vamos executar o seguinte comando para estourar o número de “inodes” permitidos para o usuário mandark:

```
1 $ touch file{1..100}
```

Depois que a quota estourou, volte ao terminal do “root”, examine o status da quota e veja se a quota do usuário mandark está também por número de arquivos:

```
1 # repquota -va
```

Quando o usuário está com a quota estourada é possível aumentar o "grace period"para ele:

```
1 # setquota -u mandark -T 86400 86400 /home
```

Onde: -u -> defini que a quota é pra um usuário mandark -> usuário que receberá a definição de quota -T -> define o príodo de grace time 86400 -> tempo em segundos (por tamanho) 86400 -> tempo em segundos (por inode) /home -> partição que será definida a quota

Voltando ao terminal logado, como usuário “mandark”, vamos apagar os arquivos criados:

```
1 $ rm a file*
```

Cheque os valores da quota do usuário mandark:

```
1 # quota -u mandark
```

Outra forma de definir a quota do usuário é através do comando setquota, este comando é muito útil para scripts.

Vamos aumentar o tamanho da quota do usuário mandark:

```
1 # setquota -u mandark 200000 210000 1000 1010 /home
```

-u -> indica que será definida quota para um usuário

mandark -> usuário que receberá os valores da quota

200000 -> soft para espaço disponível para uso

210000 -> hard para espaço disponível para uso

1000 -> soft para número de arquivos

1010 -> hard para número de arquivos

home -> partição para aplicar a quota

Cheque os valores da quota do usuário mandark:

```
# quota -u mandark
```

8.1.2 Quotas por Grupo

Defina quota por grupo para o grupo "users":

```
# setquota -g users 50000 60000 100 110
```

Ou

```
# edquota -g users
```

```
Disk quotas for group users (gid 100):
 Filesystem blocks  soft  hard  inodes  soft  hard
 /dev/sda3  0     50000   60000   0   100   110
      ^----------NÃO MEXER----------^
```

Verifique o status da quota por grupo:

```
# repquota -vag
```

Verifique os detalhes mais avançados sobre o uso das quotas nas partições.

```
# quotastats
```

Adicione o usuário mandark ao grupo users:

```
# adduser mandark users
```

Crie um diretório para teste de quota por grupo:

```
# mkdir /home/users
```

Troque sua permissão para toda vez que um arquivo for criado, pertença ao grupo users, para isso mude também o grupo do diretório para "users":

```
# chmod 2775 /home/users
# chgrp users /home/users
```

Acesse o diretório /home/users com o usuário mandark e estoure a quota de grupo por tamanho:

```
$ yes > a
```

Estoure a quota de grupo por número de arquivos:

```
$ touch arq{1..101}
```

Verifique o status da quota por grupo:

```
1 # repquota -vag
```

8.2 Replicando quotas

Verifique a quota por usuários:

```
1 # repquota -va
```

O usuário mandark tem quota definida e o usuário rh não tem, então vamos copiar a quota do usuário mandark para o usuário herdeiro:

```
1 # edquota -up mandark herdeiro
```

Onde:

-u -> usuário

-p -> protótipo

-g -> grupo

Verifique a quota por usuários:

```
1 # repquota -va
```

8.3 Criando usuário com quota definida (Só funciona no Debian)

Edite o arquivo /etc/adduser.conf e adicione um usuário que já tenha uma quota definida na opção QUOTAUSER= . Ex: o usuário mandark já tem cota definida e quero passar esta quota para um usuário novo:

```
1  # vim /etc/adduser.conf
2  # linha 67 completo com o nome do usuário que tem quota definida:
3
4  QUOTAUSER="mandark"
```

Crie um usuário com o comando adduser e veja que ele já terá quota definida:

```
1  # adduser compras
2  # repquota -va
```

8.3.1 Aviso de quota excedida

Avisos sobre quota ultrapassada podem ser enviadas automaticamente a todos os usuários pelo utilitário ‘warnquota’. Ele poderá ser executado periodicamente através do ‘cron’ (por padrão isto é feito diariamente na distribuição ’Debian’ pelo script ’/etc/cron.daily/quota’), no **CentOS** é necessário agendar a execução do comando warnquota. Dados adicionais sobre o envio das mensagens devem ser especificados no arquivo **’/etc/warnquota.conf’** seu formato é o seguinte:

```
1  # Programa usado para enviar as mensagens
2  MAIL_CMD  = "/usr/sbin/sendmail -t"
3  # Campo de origem da mensagem
```

```
FROM = "root@localhost"
# but they don't have to be:
SUBJECT = Quota excedida
CC_TO = "root@localhost"
SUPPORT = "root@localhost"
PHONE = "5555-2525"
```

O e-mail é enviado aos usuários..

Quando a quota é por grupo deve-se eleger um usuário para receber os e-mails de quota excedida do grupo.

O arquivo /etc/quotagrpadmins serve para configurar o usuário que receberá os e-mails de quota excedida do grupo:

```
# vim /etc/quotagrpadmins
grupo: usuario
users: mandark
```

Para receber o aviso para usuários e para grupos execute:

```
# warnquota -u
# warnquota -g
```

Caso queira pode colocar no crontab para executar de tempos em tempos:

```
# vim /etc/crontab
#min  hora dia_do_mes  mês dia_da_semana usuário comando
00    15     *        *       *         root    /usr/sbin/warnquota
   -u
00    15     *        *       *         root    /usr/sbin/warnquota
   -g
```

Obs.: O Debian já faz a checagem por padrão no crontab: /etc/cron.daily/quota.

4451

Linux System Administration

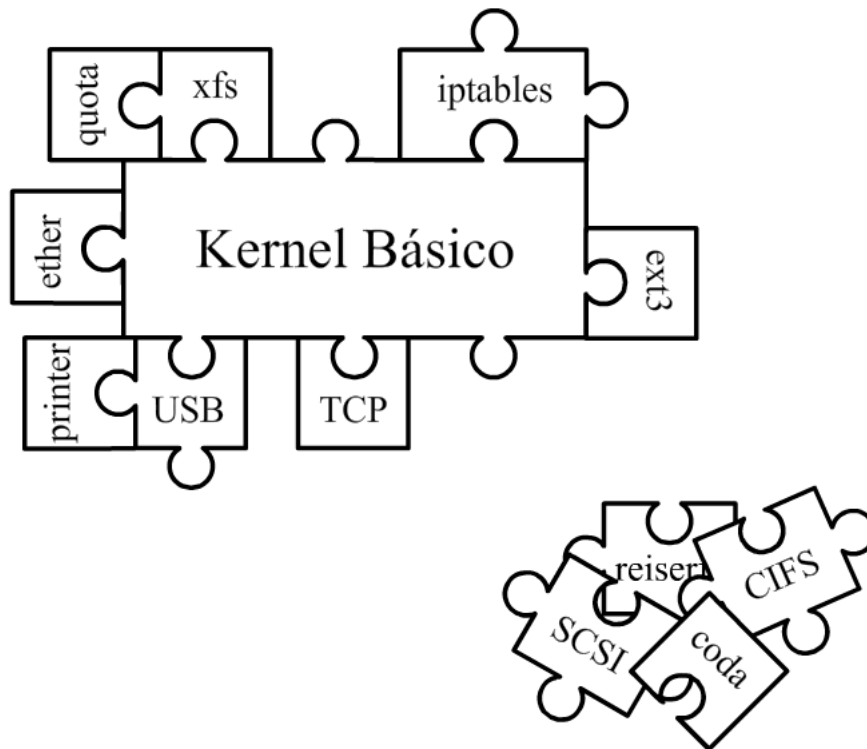
www.4linux.com.br

Conteúdo

Trabalhando com Módulos

10.1 Introdução Teórica

Quando instalamos um Debian, RedHat, Suse, Slackware, entre outras distribuições, estamos utilizando um kernel que foi compilado pelos desenvolvedores da distribuição.

O kernel que vem por padrão em uma distribuição, deve ser capaz de rodar em praticamente qualquer PC e dar suporte a quaisquer tipos de recursos que o usuário pretenda utilizar, o desenvolvedor compila um kernel que fornece todas as funcionalidades básicas e, em separado, compila pedaços de código que dão suporte a funcionalidades mais específicas. Esses pedaços de código são os chamados módulos. Dessa forma, quando o sistema é carregado, um kernel básico se coloca na memória e passa a controlar a máquina. Neste ponto são verificadas outras funcionalidades que se espera que o kernel dê suporte, como por exemplo utilizar uma partição XFS. Neste momento, se o kernel não possuir suporte nativo a esse "file system" ele irá verificar se o módulo que dá suporte a ele foi compilado e se está disponível. Se esse módulo for encontrado, ele será carregado expandindo as funcionalidades do kernel.

Em um sistema como esse, haverá diversos módulos carregados e um número maior ainda que não estará sendo utilizado, mas disponível.

Os módulos disponíveis, em geral, encontram-se no diretório “/lib/modules”, podem ser visualizados com o comando "modprobe -l"e os módulos que estão carregados podem ser visualizados com o comando “lsmod”.

Ao mesmo tempo que a capacidade de subir módulos é uma vantagem do ponto de vista que apenas os módulos realmente necessários serão carregados, há a desvantagem de fragmentação do kernel na memória.

10.2 Gerenciando os módulos

O desenvolvimento de uma nova funcionalidade para o kernel do Linux pode ser implementada diretamente no Kernel ou compilada como um módulo.

De modo geral, a escolha tende a ser a compilação como módulo. Isso se traduz no carregamento dinâmico do código apenas quando esta funcionalidade for necessária. Além disso, torna o kernel mais enxuto, leve, e portanto mais eficiente. Diversos componentes do kernel do Linux são implementados como módulos, por exemplo, filesystems, device drivers, e novas camadas de protocolos de comunicação.

Em alguns casos, compilar o código juntamente com o kernel pode ser necessário. Se um determinado componente precisa alterar alguma estrutura do kernel, ele não terá privilégios de fazer isso dinamicamente, ou mesmo que gere sua própria estrutura modificada, o restante do kernel e outros módulos carregados ainda estarão enxergando a antiga estrutura.

Ao executar o comando lsmod você pode ver quais módulos estão carregados atualmente no seu kernel. Abaixo um exemplo da saída deste comando:

```
1  # lsmod
2  Module          Size    Used    by            Tainted: P
3  ppp_generic    24060    0    (autoclean) (unused)
4  slhc            6564    0    (autoclean) [ppp_generic]
5  ircomm-tty     24224    0    (autoclean) (unused)
6  ircomm          9736    0    (autoclean) [ircomm-tty]
7  irda          112112    0    (autoclean) [ircomm-tty ircomm]
8  lp              8096    0    (autoclean)(unused)
9  parport        34176    0     (autoclean) [lp]
10 printer         8448    0    (unused)
11 agpgart        40896    3    (autoclean)
12 nvidia       1765632   11    (autoclean)
13 i810_audio     26312    0
14 soundcore       6276    0     [i810_audio]
15 ac97_codec     12488    0    [i810_audio]
16 nfsd           74256    8    (autoclean)
```

Observe a coluna Used. Ela reflete quantos processos estão fazendo uso do módulo. Todo módulo deve implementar um campo na sua estrutura de dados denominado usage counter para esta finalidade. Um módulo só pode ser removido da memória

se o seu usage counter for zero.

Os módulos são carregados através do programa insmod e uma estrutura do tipo module é alocada quando seu carregamento é solicitado. Esta estrutura contém símbolos globais que podem ser vistos pelo kernel e outros módulos, informando os pontos de entrada de suas funções, suas variáveis globais, seu usage counter, flags, entre outros.

Algumas vezes, um módulo depende de outro para realizar determinadas operações. Outro campo na estrutura module é utilizado para informar as dependências dele. Se o módulo B depende de A, este deve ser carregado antes de ser possível carregar B. O usage counter de A é incrementado sempre que um módulo que depende dele é carregado. Deste modo, não se permite que A seja removido antes de seus dependentes.

O kernel provê o comando modprobe para facilitar o gerenciamento de dependências. Este comando tenta carregar automaticamente qualquer dependência do módulo solicitado. Por exemplo, ao tentar carregar o módulo MS-DOS, o comando modprobe carrega primeiro o módulo fat, seguido por MS-DOS.

O comando modprobe faz uso de um arquivo chamado modules.dep para determinar as dependências de todos os módulos compilados para o kernel corrente. Este arquivo é gerado pela execução, no start-up da máquina, de outro programa chamado depmod. Ele avalia, durante o carregamento inicial do kernel, todos os módulos compilados, normalmente armazenados em /lib/modules, e gera o arquivo modules.dep.

Para remover um módulo da memória, utiliza-se o comando rmmod. modprobe -r ou rmmod -r são usados para remover uma pilha de módulos.

O usuário, através da configuração de um novo kernel, pode alterar o modo como diversos componentes são carregados: compilados junto com o kernel ou como módulos. Normalmente é recomendado fazer uso o máximo possível de módulos. Entretanto, em casos em que o componente é permanentemente necessário, pode ser mais eficiente compilá-lo junto com o kernel, ganhando em performance.

É de suma importância saber a versão do kernel para saber se a versão suporta ou não um módulo.

Determine qual versão do kernel está sendo utilizada:

```
1 # uname -r
```

 Dica LPI: O comando que exibe o Kernel em uso, e suas opções é: uname -a

Determine quais módulos estão carregados:

```
1 # lsmod
2 # cat /proc/modules
```

Para verificar os módulos estão carregados, usamos o comando lsmod.

Veja que a saída do comando lsmod é em colunas, é listado todos módulos que estão carregados em memória, inclusive os que não estão em uso. Onde:

Module – exibe o nome do módulo

Size – exibe em bytes, o tamanho da memória do módulo

Used by – exibe a contagem de quantas instâncias do módulo estão carregadas e o módulo que está usando; os valores são importantes porque não podemos remover um módulo que esteja sendo usado, a não ser que nesse campo, o valor seja zero. Também exibe se o módulo depende de outro para funcionar, mostrando o nome do módulo que ele depende.

Para determinar quais módulos estão compilados (disponíveis):

```
1 # modprobe -l
```

Uma das opções do comando **modprobe** é listar os módulos disponíveis em /lib/modules/. Para isso, basta utilizar a opção -l:

para determinar o número de módulos carregados e o número de módulos disponíveis:

```
1 # lsmod | grep -v  ^"Module" | wc -l
2 # modprobe -l | wc -l
```

Determine para que serve o módulo chamado ext3:

```
1 # modinfo ext3
```

O comando **modinfo** exibe informações sobre um módulo

Determine quais módulos são utilizados pelo filesystem ext3:

```
1 # lsmod | grep ext3
```

Carregue o módulo do filesystem vfat:

```
1 # modprobe vfat
```

O comando **"modprobe"** ou **"modprobe -i"** é o responsável por carregar um módulo e suas dependências. Determine quais são as dependências do módulo vfat:

```
1 # modinfo vfat
```

Determine quais módulos são utilizados pelo filesystem vfat:

```
1 # lsmod | grep vfat
```

Remova o módulo vfat:

```
1 # modprobe -r vfat
```

O comando **modprobe** também pode ser utilizado para remover módulos que não estejam sendo utilizados por outros módulos. Além dele remover o módulo, ele também remove suas dependências. Para executar essa ação, basta usar a opção -r:

Outra forma de carregar módulos é através do comando , mas diferente do comando modprobe é necessário passar o caminho completo do módulo e também é necessário carregar suas dependências primeiro.

Verifique quais são as dependências do módulo vfat:

```
1 # modinfo vfat
2 filename:       /lib/modules/2.6.32-5-686/kernel/fs/fat/vfat.ko
3 author:         Gordon Chaffee
4 description:    VFAT filesystem support
5 license:        GPL
6 srcversion:     13B4B9904275625D3971810
7 depends:        fat ,nlsbase
8 vermagic:       2.6.32-5-686 SMP mod_unload modversions 686
```

Na saída temos duas dependências para o módulo vfat, agora precisamos saber se suas dependências não são dependentes de outras dependências:

```
1 # modinfo fat
2 filename:       /lib/modules/2.6.32-5-686/kernel/fs/fat/fat.ko
3 license:        GPL
4 srcversion:     F3CEDF3D6DC8D993978847D
5 depends:        nls_base
6 vermagic:       2.6.32-5-686 SMP mod_unload modversions 686
```

Verificado que o módulo fat depende do módulo nls_base. Agora verifique se o módulo nls_base não depende de outro módulo:

```
1 # modinfo nls_base
2 filename:       /lib/modules/2.6.32-5-686/kernel/fs/nls/nls_base.ko
3 license:        DUAL BSD/GPL
4 depends:
5 vermagic:       2.6.32-5-686 SMP mod_unload modversions 686
```

O módulo nls_base não depende de nenhum outro módulo, então agora, carregue os módulos na ordem de dependências:

nls_base -> fat -> vfat

Lembrando que você deve passar o caminho completo do módulo para utilizar o comando .

Para visualizar todos os módulos disponíveis utiliza-se o comando “modprobe” com a opção “-l”, mas para visualizar se um módulo específico está disponível utilize seu nome como argumento.

Todos os módulos:

```
1 # modprobe -l
```

Módulo específico:

```
1 # modprobe -l nls_base
2 # modprobe -l fat
3 # modprobe -l vfat
```

Então para carregar o módulo nls_base fat e vfat faça:

```
1 # insmod /lib/modules/$(uname -r)/$(modprobe -l nls_base)
2 # insmod /lib/modules/$(uname -r)/$(modprobe -l fat)
3 # insmod /lib/modules/$(uname -r)/$(modprobe -l vfat)
```

Veja que todos eles foram carregados com sucesso:

```
1 # lsmod | grep fat
```

Outra forma de remover módulos é através do comando **rmmod**, mas diferente do comando “modprobe -r” ele só remove o módulo que não esteja sendo utilizado por outro e não remove suas dependências.

Na saída do comando anterior verifique qual módulo está sendo utilizado por outro:

vfat -> não tem dependente

fat -> vfat depende dele

nls_base -> fat depende dele

Logo para removermos temos que seguir a ordem:

vfat -> fat -> nls_base

Removendo os módulos:

```
# rmmod vfat
# rmmod fat
# rmmod nls_base
```

Veja que todos eles foram descarregados com sucesso:

```
# lsmod | grep fat
```

Como o "modprobe" sabe quais módulos dependem de quais módulos?

```
# cd /lib/modules/$(uname -r)
# ls -l
# less modules.dep
```

O arquivo modules.dep é o responsável por armazenar os dados de dependências de módulos, através dele os comandos modprobe e modinfo, conseguem obter as informações necessárias para serem executados, teste arquivo é gerado em todo boot.

Não acredita que o "modprobe" usa esse arquivo? Remova-o e tente carregar o módulo vfat:

```
# rm /lib/modules/$(uname -r)/modules.dep
# modprobe vfat
```

Não funcionou? E agora? Construa o arquivo “modules.dep” e tente novamente:

```
1 # depmod
2 # ls /lib/modules/$(uname -r)/modules.dep
3 # modprobe vfat
```

O comando **depmod** gera o arquivo modules.dep.

10.2.1 Identificando Dispositivos e seus módulos

Identifique qual é a placa de rede do seu computador:

```
1 # lspci -nn | grep -i eth
2 03:00.0 Ethernet controller [0200]: Realtek Semiconductor Co., Ltd.
    RTL8101E/RTL8102E PCI Express Fast Ethernet controller [10ec
    :8136] (rev 02)
```

DICA: Repare no número **10ec:8136 (PCI iD)** este número é único para este dispositivo e através dele podemos saber qual o nome do seu módulo e qual versão do kernel tem suporte a ele.

 Acesse o site: http://www.kmuto.jp/debian/hcl/

Digite o número encontrado: **10ec:8136** e descubra qual o nome do módulo e quais versões do kernel têm suporte a este módulo.

Outra forma de identificar o modulo, de um dispositivo é o "lspci", com filtro de uma palavra chave, continuando o exemplo da placa de rede:

```
1 # lspci | grep -i net
2 04:00.0 Ethernet controller: Marvell Technology Group Ltd. 88E8057
     PCI-E Gigabit Ethernet Controller (rev 10)
```

Repare o número de id da placa de rede **04:00.0** e descubra qual o nome do módulo que a sua placa de rede necessita:

```
1 # lspci -v -s 04:00.0
2 04:00.0 Ethernet controller: Marvell Technology Group Ltd. 88E8057
     PCI-E Gigabit Ethernet Controller (rev 10)
3   Subsystem: Sony Corporation Device 907a
4   Flags: bus master, fast devsel, latency 0, IRQ 46
5   Memory at e6620000 (64-bit, non-prefetchable) [size=16K]
6   I/O ports at a000 [size=256]
7   Expansion ROM at e6600000 [disabled] [size=128K]
8   Capabilities: [48] Power Management version 3
9   Capabilities: [5c] MSI: Enable+ Count=1/1 Maskable- 64bit+
10  Capabilities: [c0] Express Legacy Endpoint, MSI 00
11  Capabilities: [100] Advanced Error Reporting
12  Capabilities: [130] Device Serial Number 6b-3b-74-ff-ff-49-42-54
13  Kernel driver in use: sky2
14  Kernel modules: sky2
```

Outra forma de busca seria buscar informações na internet ou tentar determinar qual é o módulo que ela utiliza na raça(tentativa e erro):

```
1 # modprobe -l |grep -i realtek
2 # modprobe -l |grep -i real
3 # modprobe -l |grep -i tek
4 # modprobe -l |grep rtl
5 # modprobe ...(módulos obtidos nas sáidas)
```

Não há uma regra geral para determinar qual é o módulo que fornece suporte a um determinado hardware. A forma mais fácil é utilizar um kernel genérico e tentar descobrir qual é o módulo que é utilizado por meio dos comandos "lsmod" e "modinfo", procurar na árvore do kernel, ou procurar nos mecanismos de busca na Internet.

Após descobrir o módulo descarregue-o e veja que a placa rede parou de funcionar:

```
1 # modprobe -r r8169
```

No CentOS ao derrubar o módulo da placa de rede ele recarrega o módulo automaticamente, isto porque existe o arquivo: "/etc/sysconfig/network-scripts/network-functions", este arquivo mantém funções que controlam muitos scripts de interface e funcões que estão em contato com programas em execução que tenham solicitado informações sobre mudanças no status de uma interface.

Caso queira bloquear o carregamento de um módulo na inicialização edite o arquivo "/etc/modprobe.d/blacklist.conf" e adicione blacklist [módulo].

Bloqueie o módulo da placa de rede de ser carregado na hora do boot:

```
1 # vim /etc/modprobe.d/blacklist.conf
2 blacklist r8169
```

Reinicie a máquina e veja que o módulo não é carregado:

```
1 # lsmod | grep r8169
```

Caso precise que um módulo seja carregado automaticamente na hora do boot faça:

Debian:

```
# vim /etc/modules
zaurus
```

CentOS:

```
# vim /etc/rc.modules
modprobe zaurus
# chmod +x /etc/rc.modules
```

Verifique que o módulo zaurus para “PDA ZAURUS” não está carregado, reinicie a máquina e veja que ele será carregado automaticamente, após o boot:

```
# lsmod | grep zaurus
```

Servidor de Impressão

10.3 Introdução Teórica

O CUPS - Common Unix Printing System é uma das formas mais utilizadas atualmente para trabalhar com impressão no mundo GNU/Linux. Ele utiliza o protocolo IPP - Internet Printing Protocol para gerenciar as filas e trabalhos de impressão. Com o "IPP" você pode imprimir de qualquer lugar, através da internet para sua impressora doméstica, por exemplo. Além disso, o "CUPS" fornece uma interface "Web" para gerenciamento de quotas de impressão e que oferece suporte à maioria das impressoras existentes.

A melhor documentação a respeito do CUPS pode ser encontrada online no manual oficial no projeto, disponível no endereço "http://www.cups.org"

10.3.1 Instalação Servidor Cups

Debian:

```
1 # apt-get install cups-bsd
```

CentOS:

```
1 # yum install cups-lpd
```

Agora podemos gerenciar as configurações relacionadas à impressora. Para isso utilizaremos a interface de gerenciamento do CUPS via “browser”. Para isso, abra seu navegador preferido e digite “http://localhost:631”.

Antes de continuarmos a instalação do servidor de impressão devemos saber se a impressora tem suporte ao linux e qual é o seu driver.

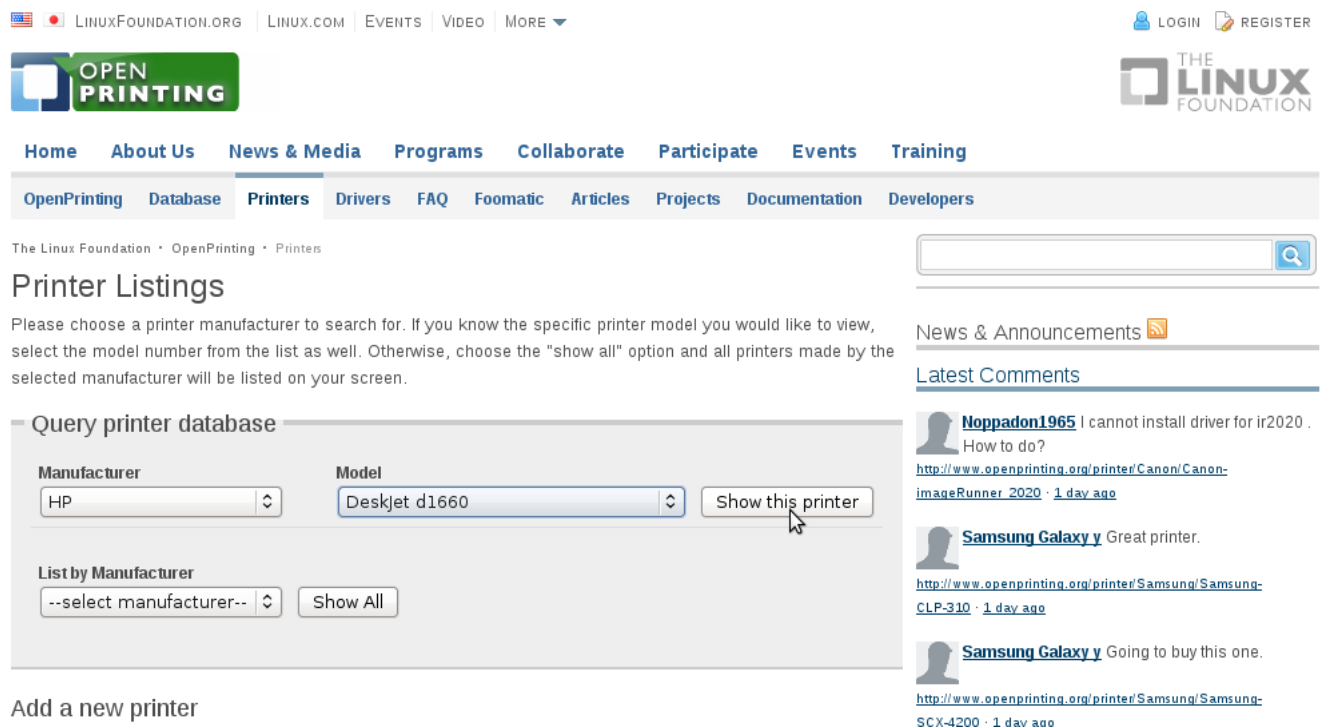
Para isso identifique o nome e modelo da impressora, no nosso exemplo a impressora será a “HP Deskjet D1660”.

DICA: Para descobrir o driver da impressora e se ela tem suporte acesse o site: http://www.openprinting.org

Clique em Printers:

We have switched over to a new generation of web pages to browse and manage our printer/driver database. The pages are not only looking nicer and better fitting into the general web site of the Linux Foundation, they give also much quicker access by being backed by a relationale database and two mirrored servers and we will not get so many unuseful printer entry contributions any more as login with a Linux Foundation account is required for contributors now.

In addition we have now facilities for driver developers and printer manufacturers to easily contribute driver and printer entries via a web interface.

More info. Start browsing: Printers, Drivers

Enjoy the new OpenPrinting database web pages!

For Developers

The goal of the OpenPrinting workgroup is to develop and promote a set of standards that will address the complete printing needs of embedded, mobile, desktop, enterprise, and production environments, including management, reliability, security, scalability, printer feature access and network accessibility. This is achieved by

- creating a Common Printing Dialog for all applications and desktops
- developing standard APIs for printing
- collecting information about printers and printer drivers and providing the drivers in distribution-independent packages
- integrating them in existing operating systems

Selecione o fabricante(Manufacter) e modelo(Model) e depois clique em “Show this printer”:

Aparecerá o nome da impressora no topo da página:

E no final aparecerá o nome do driver, selecione-o:

Selecione novamente o driver:

Antes de fazer o download, pode-se tentar instalar o driver “hplip” pelo gerenciador de pacotes de sua distribuição:

Debian:

```
1 # apt-get install hplip
```

Verifique a versão:

```
1 # dpkg -l hplip
```

CentOS:

```
1 # yum install hplip
```

Verifique a versão:

```
1 # rpm -q hplip
```

Compare as versões com a do site, talvez sua versão não tenha suporte para a impressora, então faça o download da versão mais atual:

Ao clicar em download, aparecerá a tela abaixo, preencha conforme o solicitado: No CentOS:

Clique em avançar, verfique que na próxima tela, é apresentado que o pacote fornecido pela distribuição não tem suporte a esta impressora: “Red Hat Enterprise linux 6.0 supplies HPLIP 1.6.7 and it does not support your printer”.

Installation Wizard

You have selected Red Hat Enterprise Linux 6.0 using the HP Deskjet d1660 Printer.

Red Hat Enterprise Linux 6.0 supplies HPLIP 1.6.7 and it does not support your printer.

You must download and install HPLIP in order to use your printer with Red Hat Enterprise Linux 6.0.

Please click *Previous* to select a different operating system or printer.

Please click *Cancel* to return to the HPLIP home page.

Cancel »

Clique em avançar:

If you do not see your processor architecture below, click *Download Installer* to download the HPLIP installer.

Download Installer »

Step 6: Select Processor

x86_64

Step 7: License Agreement

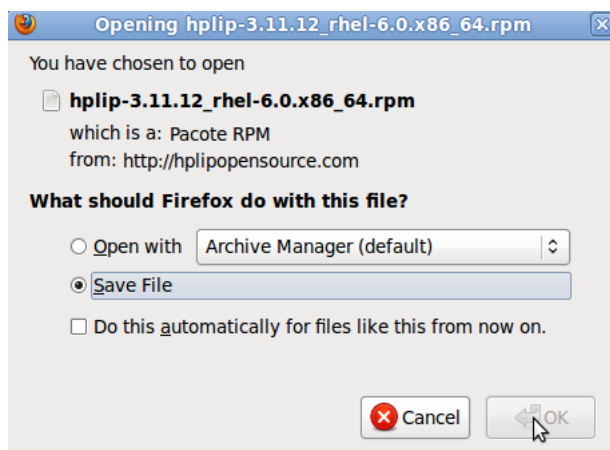
Before downloading the HPLIP package, you must read and agree with the terms of the following license agreement.

I agree to the terms and conditions of the license agreement.

Download

Please click *Restart* to select a different operating system or printer.

Restart »

Faça o download:

Para instalar:

```
1 # rpm -i hplip-3.11.12_rhel-6.0.x86_64.rpm
```

No Debian:

Na tela seguinte é alertado que o Debian6 fornece uma versão do driver hplip que não suporte a impressora HP. "Debian 6.0 supplies HPLIP 2.8.6 and it does not support

your printer”

Clique em avançar:

Clique em avançar novamente:

Para finalizar faça o download do driver:

Installer Walkthrough

Step 1: Download the Automatic Installer (.run file)

Download HPLIP 3.11.10:

(Download Digital Certificate) What's this?

Step 2: Run the Automatic Installer

Running the installer requires that you open a command shell to enter commands. To do this, open a terminal or console window (how do I open a terminal?).

In the terminal/console, enter the following commands (type all the text after the $ character and then press enter):

Execute o arquivo baixado:

```
# bash hplip-3.11.10.run
Creating directory hplip-3.11.10
Verifying archive integrity... All good.
Uncompressing HPLIP 3.11.10 Self Extracting Archive
   ..........................................................................

warning: hplip-install should not be run as root.
HP Linux Imaging and Printing System (ver. 3.11.10)
HPLIP Installer ver. 5.1

Copyright (c) 2001-9 Hewlett-Packard Development Company, LP
This software comes with ABSOLUTELY NO WARRANTY.
This is free software, and you are welcome to distribute it
under certain conditions. See COPYING file for more details.

Installer log saved in: hplip-install_Mon-21-Nov-2011_17:18:45.log

/error: You are running the installer as root. It is highly
   recommended that you run the installer as
error: a regular (non-root) user. Do you still wish to continue?
Continue instalando:
                                                      Continue with
    installation (y=yes, n=no*, q=quit) ? y

note: Defaults for each question are maked with a '*'. Press <enter>
    to accept the default.
```

```
21
22
23 INSTALLATION MODE
24 -----------------
25 Automatic mode will install the full HPLIP solution with the most
      common options.
26 Custom mode allows you to choose installation options to fit
      specific requirements.
27 Qual o tipo de instalação: automática
28 Please choose the installation mode (a=automatic*, c=custom, q=quit)
       : a
29 Initializing. Please wait...
30
31 INTRODUCTION
32 ------------
33 This installer will install HPLIP version 3.11.10 on your computer.
34 Please close any running package management systems now (YaST, Adept
      , Synaptic, Up2date, etc).
35
36
37 DISTRO/OS CONFIRMATION
38 ----------------------
39 Distro appears to be Debian 6.0.3.
40 DICA:Quando perguntar qual a versão da sua distro no cado do Debian,
       diga que não está correto e selecione manualmente a versão 6.0,
      o pacote não é atualizado constantemente quanto a distribuição e
      por isso, falhará a instalação postriormente se não escolher
      manualmente.
41 Is "Debian 6.0.3" your correct distro/OS and version (y=yes*, n=no,
      q=quit) ? n
42
43 DISTRO/OS SELECTION
44 -------------------
45
46 Choose the name of the distro/OS that most closely matches your
      system:
47
```

```
48 Num. Distro/OS Name
49 ---- ------------------------
50 0 Mepis
51 1 Debian
52 2 SUSE Linux
53 3 Mandriva Linux
54 4 Fedora
55 5 Red Hat
56 6 Red Hat Enterprise Linux
57 7 Ubuntu
58 8 PCLinuxOS
59 9 Linux Mint
60 10 gOS
61 11 Linpus Linux
62 12 IGOS
63 13 Boss
64 14 Linux From Scratch
65 Escolha Debian: 1
66 Enter number 0...14 (q=quit) ? 1
67
68 Choose the version of "Debian" that most closely matches your system
   :
69
70 Num. Distro/OS Version
71 --------------------------------------------
72 0  Unknown or not listed
73 1  5.0 ("Lenny")
74 2  5.0.1 ("Lenny")
75 3  5.0.2 ("Lenny")
76 4  5.0.3 ("Lenny")
77 5  5.0.4 ("Lenny")
78 6  5.0.5 ("Lenny")
79 7  5.0.6 ("Lenny")
80 8  5.0.7 ("Lenny")
81 9  5.0.8 ("Lenny")
82 10 6.0 ("Squeeze")
83 11 6.0.1 ("Squeeze")
```

```
84  12 6.0.2 ("Squeeze")
85  Escolha a versão mais próxima a sua distro:
86  Enter number 0...12 (q=quit) ? 12
87
88  Distro set to: Debian 6.0.2
89
90
91  INSTALLATION NOTES
92  ------------------
93  NOTE: Disable the CD Sources in your apt sources.list or the install
        will fail and hang.
94
95  Please read the installation notes. Press <enter> to continue or 'q'
        to quit:
96
97
98  RUNNING PRE-INSTALL COMMANDS
99  ----------------------------
100 OK
101
102
103 INSTALL MISSING REQUIRED DEPENDENCIES
104 -------------------------------------
105 warning: There are 8 missing REQUIRED dependencies.
106 note: Installation of dependencies requires an active internet
      connection.
107 warning: Missing REQUIRED dependency: gcc (gcc - GNU Project C and C
      ++ Compiler)
108 warning: Missing REQUIRED dependency: make (make - GNU make utility
      to maintain groups of programs)
109 warning: Missing REQUIRED dependency: python-devel (Python devel -
      Python development files)
110 warning: Missing REQUIRED dependency: cups-devel (CUPS devel- Common
        Unix Printing System development files)
111 warning: Missing REQUIRED dependency: libusb (libusb - USB library)
112 warning: Missing REQUIRED dependency: libtool (libtool - Library
      building support services)
```

```
113 warning: Missing REQUIRED dependency: cups-image (CUPS image - CUPS
       image development files)
114 warning: Missing REQUIRED dependency: libjpeg (libjpeg - JPEG
       library)
115
116
117 INSTALL MISSING OPTIONAL DEPENDENCIES
118 -------------------------------------
119 warning: There are 10 missing OPTIONAL dependencies.
120 note: Installation of dependencies requires an active internet
       connection.
121 warning: Missing REQUIRED dependency for option 'network': libcrypto
        (libcrypto - OpenSSL cryptographic library)
122 warning: Missing REQUIRED dependency for option 'network':
       libnetsnmp-devel (libnetsnmp-devel - SNMP networking library
       development files)
123 warning: Missing REQUIRED dependency for option 'gui_qt4': pyqt4-
       dbus (PyQt 4 DBus - DBus Support for PyQt4)
124 warning: Missing REQUIRED dependency for option 'gui_qt4': pyqt4 (
       PyQt 4- Qt interface for Python (for Qt version 4.x))
125 warning: Missing OPTIONAL dependency for option 'fax': reportlab (
       Reportlab - PDF library for Python)
126 warning: Missing REQUIRED dependency for option 'fax': dbus (DBus -
       Message bus system)
127 warning: Missing REQUIRED dependency for option 'scan': sane-devel (
       SANE - Scanning library development files)
128 warning: Missing OPTIONAL dependency for option 'scan': pil (PIL -
       Python Imaging Library (required for commandline scanning with hp
       -scan))
129 warning: Missing OPTIONAL dependency for option 'scan': xsane (xsane
        - Graphical scanner frontend for SANE)
130 warning: Missing OPTIONAL dependency for option 'base': cups-ddk (
       CUPS DDK - CUPS driver development kit)
131 warning: This installer cannot install 'cups-ddk' for your distro/OS
        and/or version.
132
133
```

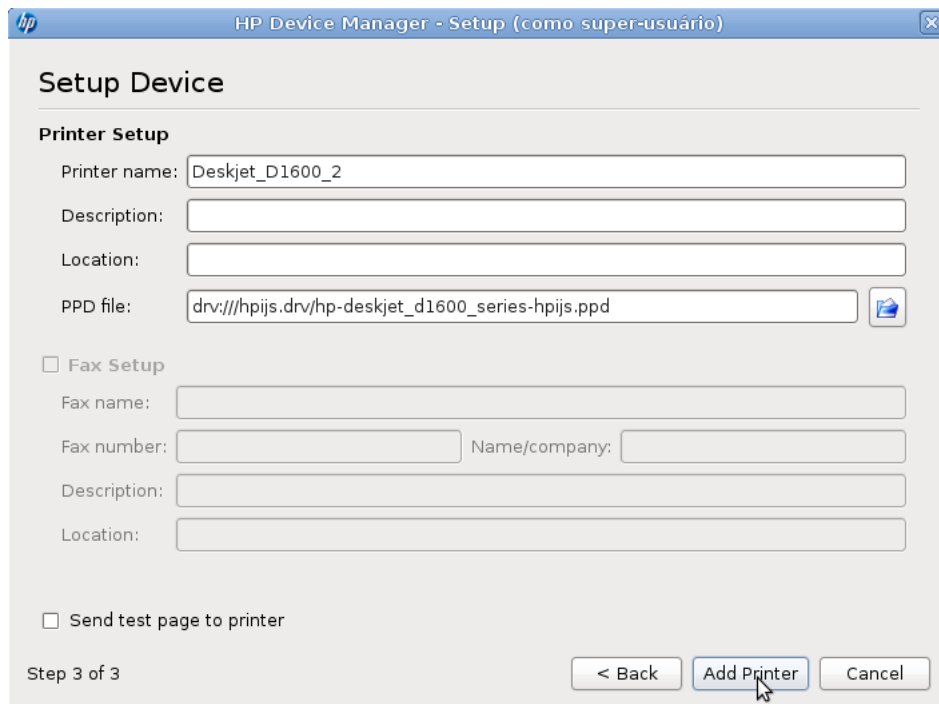
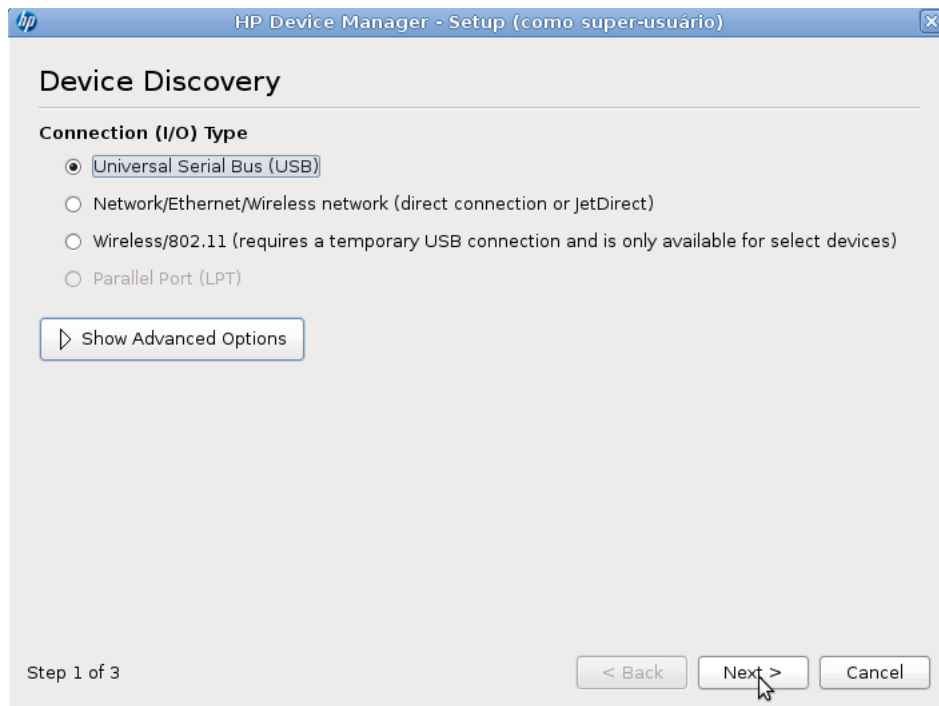
```
134 CHECKING FOR NETWORK CONNECTION
135 -------------------------------
136 Network connection present.
137
138
139 RUNNING PRE-PACKAGE COMMANDS
140 ----------------------------
141 su -c "dpkg --configure -a" (Pre-depend step 1)
142 su -c "apt-get install -f" (Pre-depend step 2)
143 su -c "apt-get update" (Pre-depend step 3)
144 warning: An error occurred running 'su -c "apt-get install --yes
       cupsys-bsd"'
145 su -c "apt-get install --yes cupsys-bsd" (Pre-depend step 4)
146 OK
147
148
149 DEPENDENCY AND CONFLICT RESOLUTION
150 ----------------------------------
151 Running 'su -c "apt-get install --force-yes -y g++"'
152 Please wait, this may take several minutes...
153 Running 'su -c "apt-get install --force-yes -y make"'
154 Please wait, this may take several minutes...
155 Running 'su -c "apt-get install --force-yes -y python-dev"'
156 Please wait, this may take several minutes...
157 Running 'su -c "apt-get install --force-yes -y libcups2-dev"'
158 Please wait, this may take several minutes...
159 Running 'su -c "apt-get install --force-yes -y cups-bsd"'
160 Please wait, this may take several minutes...
161 Running 'su -c "apt-get install --force-yes -y cups-client"'
162 Please wait, this may take several minutes...
163 Running 'su -c "apt-get install --force-yes -y libusb-dev"'
164 Please wait, this may take several minutes...
165 Running 'su -c "apt-get install --force-yes -y libtool"'
166 Please wait, this may take several minutes...
167 Running 'su -c "apt-get install --force-yes -y libcupsimage2"'
168 Please wait, this may take several minutes...
169 Running 'su -c "apt-get install --force-yes -y libcupsimage2-dev"'
```

```
170 Please wait, this may take several minutes...
171 Running 'su -c "apt-get install --force-yes -y libjpeg62-dev"'
172 Please wait, this may take several minutes...
173 Running 'su -c "apt-get install --force-yes -y libssl-dev"'
174 Please wait, this may take several minutes...
175 Running 'su -c "apt-get install --force-yes -y libsnmp-dev"'
176 Please wait, this may take several minutes...
177 Running 'su -c "apt-get install --force-yes -y python-qt4-dbus"'
178 Please wait, this may take several minutes...
179 Running 'su -c "apt-get install --force-yes -y python-qt4"'
180 Please wait, this may take several minutes...
181 Running 'su -c "apt-get install --force-yes -y python-reportlab"'
182 Please wait, this may take several minutes...
183 Running 'su -c "apt-get install --force-yes -y libdbus-1-dev"'
184 Please wait, this may take several minutes...
185 Running 'su -c "apt-get install --force-yes -y libsane-dev"'
186 Please wait, this may take several minutes...
187 Running 'su -c "apt-get install --force-yes -y python-imaging"'
188 Please wait, this may take several minutes...
189 Running 'su -c "apt-get install --force-yes -y xsane"'
190 Please wait, this may take several minutes...
191
192 RUNNING POST-PACKAGE COMMANDS
193 -----------------------------
194 OK
195
196
197 RE-CHECKING DEPENDENCIES
198 ------------------------
199 warning: An optional dependency 'pil (PIL - Python Imaging Library (
       required for commandline scanning with hp-scan))' is still
       missing.
200 warning: Some features may not function as expected.
201
202
203 PRE-BUILD COMMANDS
204 ------------------
```

```
205 OK
206
207
208 BUILD AND INSTALL
209 -----------------
210 Running './configure --with-hpppddir=/usr/share/ppd/HP --libdir=/usr
       /lib64 --prefix=/usr --enable-qt4 --enable-doc-build --disable-
       cups-ppd-install --disable-foomatic-drv-install --disable-
       foomatic-ppd-install --disable-hpijs-install --disable-policykit
       --enable-cups-drv-install --enable-hpcups-install --enable-
       network-build --enable-dbus-build --enable-scan-build --enable-
       fax-build'
211 Please wait, this may take several minutes...
212 Command completed successfully.
213
214 Running 'make clean'
215 Please wait, this may take several minutes...
216 Command completed successfully.
217
218 Running 'make'
219 Please wait, this may take several minutes...
220 Command completed successfully.
221
222 Running 'make install'
223 Please wait, this may take several minutes...
224 Command completed successfully.
225
226
227 Build complete.
228
229
230 POST-BUILD COMMANDS
231 -------------------
232 /usr/sbin/usermod -a -G lp,lpadmin root (Post-build step 1)
233
234
235 RESTART OR RE-PLUG IS REQUIRED
```

```
236 ------------------------------
237 If you are installing a USB connected printer, and the printer was
        plugged in
238 when you started this installer, you will need to either restart
        your PC or
239 unplug and re-plug in your printer (USB cable only). If you choose
        to restart,
240 run this command after restarting: hp-setup (Note: If you are using
        a parallel
241 connection, you will have to restart your PC. If you are using
        network/wireless,
242 you can ignore and continue).
243 Restart or re-plug in your printer (r=restart, p=re-plug in*, i=
        ignore/continue, q=quit) :
244 Please unplug and re-plugin your printer now. Press <enter> to
        continue or 'q' to quit: <enter>
```

Com a inmpressora ligada e conectada ao servidor: execute o comando instalado pelo pacote:

```
1 # hp-setup
```

Escolha o tipo de conexão da impressora e avance:

O dispositivo será reconhecido:

Adicione a impressora:

Pode-se adicionar a impressora diretamente pela sua interface gráfica caso o driver já esteja instalado:

Acesse: http://localhost:631

Preencha o nome da impressora, os demais campos não são obrigatórios, este será o nome que aparecerá na rede:

Caso sua impressora já esteja conectada ao servidor e ligada, ela será detectada automaticamente, escolha a forma com que ela está conectada corretamente, caso ela não tenha sido reconhecida.

Escolha o driver para o tipo da sua impressora:

Digite a senha do administrador para adicionar a impressora, todos que pertençam ao grupo lpadmin são administradores do cups.

Imprima uma página teste:

10.4 Compartilhando a impressora

Na aba Administration do lado direito, habilite:

- Visualizar impressoras compartilhadas por outros sistemas

- Compartilhar impressoras conectadas a este servidor

- Habilitar administração remota

Tudo o que pode ser feito no gerenciador gráfico, pode ser feito no arquivo de configuração do servidor cups:

```
1 # cat /etc/cups/cupsd.conf
```

10.5 Configuração do Cliente

O arquivo para configuração do client é o client.conf que por padrão não existe ou está vazio:

```
1 # vim /etc/cups/client.conf
```

Esse arquivo deve conter o endereço IP do Servidor de Impressão na rede:

ServerName 192.168.200.254

Reinicie o serviço do cups:

```
1 # /etc/init.d/cups restart
```

Definimos nossa impressora conectada através da porta “USB”, mas e se fosse uma impressora da rede, ou Paralela? Podemos conferir como ela seria referenciada usando o comando abaixo:

```
1 # lpinfo -v
```

Dica LPI: Se você tivesse um impressora em na segunda porta paralela, esta seria referenciada como “/dev/lp1”.

Podemos agora fazer um teste de impressão com o comando “lp”:

```
1 # lp -dnome_da_impressora /etc/shadow
```

Dica LPI: a opção -d"recebe o nome da minha impressora /etc/shadow é o arquivo que será impresso

.

Verifique agora a fila de impressão atual:

```
1 # lpstat -t
```

ou

```
1 # lpq -Pnome_da_empresa
```

Para remover o trabalho da fila de impressão por modo texto execute:

```
1 # lprm -Pnome_da_impressora número_do_job
```

4451

Linux System Administration

www.4linux.com.br

Conteúdo

Redes avançado

11.1 Introdução Teórica

11.1.1 IPV4

No IPV4, os endereço IP são compostos por 4 blocos de 8 bits (32 bits no total), que são representados através de números de 0 a 255, como "200.156.23.43"ou "64.245.32.11".

As faixas de endereços começadas com **"10"**, com **"192.168"** ou com de **"172.16"** até **"172.31"** são reservadas para uso em redes locais e por isso não são usados na internet. Os roteadores que compõe a grande rede são configurados para ignorar estes pacotes, de forma que as inúmeras redes locais que utilizam endereços na faixa "192.168.0.x"(por exemplo) podem conviver pacificamente.

Embora aparentem ser uma coisa só, os endereços IP incluem duas informações. O endereço da rede e o endereço do host dentro dela. Em uma rede doméstica, por exemplo, você poderia utilizar os endereços "192.168.1.1", "192.168.1.2"e "192.168.1.3", onde o "192.168.1."é o endereço da rede (e por isso não muda) e o último número (1, 2 e 3) identifica os três micros que fazem parte dela.

Os micros da rede local podem acessar a internet através de um roteador, que pode ser tanto um servidor com duas placas de rede, quando um modem ADSL ou outro dispositivo que ofereça a opção de compartilhar a conexão. Neste caso, o roteador passa a ser o gateway da rede e utiliza seu endereço IP válido para encaminhar as

requisições feitas pelos micros da rede interna. Este recurso é chamado de NAT (Network Address Translation).

Endereços de 32 bits permitem cerca de 4 bilhões de endereços diferentes, quase o suficiente para dar um endereço IP exclusivo para cada habitante do planeta. Os endereços são divididos em:

Classe	*Valor do primeiro octeto*	*Especificação dos octetos*
A	0 até 127	rede.HOST.HOST.HOST
B	128 até 191	rede.rede.HOST.HOST
C	192 até 223	rede.rede.rede.HOST
D	224 até 239	-
E	240 até 255	-

O grande problema é que os endereços são sempre divididos em duas partes, rede e host. Nos endereços de classe A, o primeiro octeto se refere à rede e os três octetos seguintes referem-se ao host. Temos apenas 126 faixas de endereços classe A disponíveis no mundo, dadas a governos, instituições e até mesmo algumas empresas privadas, como por exemplo a IBM. As faixas de endereços classe A consomem cerca de metade dos endereços IP disponíveis, representando um gigantesco desperdício, já que nenhuma das faixas é completamente utilizada. Será que a IBM utiliza todos os 16 milhões de endereços IP a que tem direito? Certamente não.

Mesmo nos endereços classe B (dois octetos para a rede, dois para o host, garantindo 65 mil endereços) e nos classe C (três octetos para a rede e um para o host, ou seja, apenas 256 endereços) o desperdício é muito grande. Muitas empresas alugam faixas de endereços classe C para utilizar apenas dois ou três endereços por exemplo.

Para piorar, parte dos endereços estão reservados para as classes D e E, que jamais foram implementadas. Isto faz com que já haja uma grande falta de endereços, principalmente os de classe A e B, que já estão todos ocupados.

11.2 Máscara de rede

A máscara de rede, juntamente com o endereço IP, define a rede o computador pertence, isto é, que outros endereços IP que o computador pode comunicar diretamente na mesma LAN.

A fim de compreender a máscara lembre-se sempre que os 4 bytes que define tanto o endereço IP e a máscara de rede poderiam ser representados em formato binário.

A máscara de rede é, por definição, uma seqüência de "1"a partir da esquerda para a direita, seguido por um certo número de "0"(a faixa de rede). Devido a esta regra a máscara de rede é muitas vezes representada com valores decimais, que soam como um ou mais "255", seguido por um ou mais "0".

Utilizando máscaras de sub-rede padrão para cada classe de endereços, onde são utilizados oito, dezesseis ou vinte e quatro bits para a máscara de rede, conforme descrito a seguir:

Classes	Máscara padrão	
A=8bits	255.0.0.0 rede.host.host.host	
B=16bits	255.255.0.0 rede.rede.host.host	
C=24bits	255.255.255.0 rede.rede.rede.host	

Por isso que existe uma outra notação conhecida como CIDR (Classless Inter-Domain Routing), onde a máscara de sub-rede é indicada simplesmente pelo número de bits utilizados na máscara de sub-rede, conforme exemplos a seguir:

Definição da rede	Máscara de rede
192.168.0.0.0/8	255.255.255.0
172.16.0.0.0/24	255.255.0.0
10.0.0.0/16	255.0.0.0

Porém com este esquema de endereçamento, baseado apenas nas máscaras de rede padrão para cada classe (oito, dezesseis ou vinte e quatro bits), haveria um grande desperdício de números IP. Por exemplo, que empresa no mundo precisaria da faixa completa de uma rede classe A, na qual estão disponíveis mais de 16 milhões de endereços IP?

Analise o outro extremo desta questão. Imagine, por exemplo, uma empresa de porte médio, que tem a matriz em São Paulo e mais cinco filiais em outras cidades do Brasil. Agora imagine que em nenhuma das localidades, a rede tem mais do que 30 computadores. Se for usado as máscaras de sub-rede padrão, teria que ser definida uma rede Classe C (até 254 computadores), para cada localidade. Observe que estamos reservando 254 números IP para cada localidade (uma rede classe C com máscara 255.255.255.0), quando na verdade, no máximo, 30 números serão utilizados em cada localidade. Na prática, um belo desperdício de endereços IP, mesmo em um empresa de porte médio ou pequeno.

Observe que neste exemplo, uma única rede Classe C seria suficiente. Já que são seis localidades (a matriz mais seis filiais), com um máximo de 30 endereços por localidade, um total de 254 endereços de uma rede Classe C seria mais do que suficiente. Ainda haveria desperdício, mas agora bem menor.

A boa notícia é que é possível "dividir" uma rede (qualquer rede) em sub-redes, onde cada sub-rede fica apenas com uma faixa de números IP de toda a faixa original. Por exemplo, a rede Classe C 200.100.100.0/255.255.255.0, com 256 números IPs disponíveis (na prática são 254 números que podem ser utilizados, descontando o primeiro que é o número da própria rede e o último que o endereço de broadcast, poderia ser dividida em 8 sub-redes, com 32 números IP em cada sub-rede. O esquema a seguir ilustra este conceito:

Rede original: 256 endereços IP disponíveis: 200.100.100.0 -> 200.100.100.255 Divisão da rede em 8 sub-redes, onde cada sub-rede fica com 32 endereços IP:

Sub-rede 01: 200.100.100.0 -> 200.100.100.31

Sub-rede 02: 200.100.100.32 -> 200.100.100.63

Sub-rede 03: 200.100.100.64 -> 200.100.100.95

Sub-rede 04: 200.100.100.96 -> 200.100.100.127

Sub-rede 05: 200.100.100.128 -> 200.100.100.159

Sub-rede 06: 200.100.100.160 -> 200.100.100.191

Sub-rede 07: 200.100.100.192 -> 200.100.100.223

Sub-rede 08: 200.100.100.224 -> 200.100.100.255

Para o exemplo da empresa com seis localidades (matriz mais cinco filiais), onde, no máximo, são necessários trinta endereços IP por localidade, a utilização de uma única rede classe C, dividida em 8 sub-redes seria a solução ideal. Na prática a primeira e a última sub-rede são descartadas, pois o primeiro IP da primeira sub-rede representa o endereço de rede e o último IP da última sub-rede representa o endereço de broadcast. Com isso restariam, ainda, seis sub-redes. Exatamente a quantia necessária para o exemplo proposto. Observe que ao invés de seis redes classe C, bastou uma única rede Classe C, subdividida em seis sub-redes. Uma bela economia de endereços. Claro que se um dos escritórios, ou a matriz, precisasse de mais de 32 endereços IP, um esquema diferente de divisão teria que ser criado.

Entendido o conceito teórico de divisão em sub-redes, resta o trabalho prático, ou seja:

- O que tem que ser alterado para fazer a divisão em sub-redes?

- Como calcular o número de sub-redes e o número de números IP dentro de cada sub-rede?

- Como listar as faixas de endereços dentro de cada sub-rede?

Observe o que tem que ser alterado para fazer a divisão de uma rede padrão (com máscara de 8, 16 ou 24 bits) em uma ou mais sub-redes. Em seguida, veja alguns exemplos de divisão de uma rede em sub-redes. Mãos a obra.

11.2.1 Alterando o número de bits da máscara de sub-rede

Por padrão são utilizadas máscaras de sub-rede de 8, 16 ou 24 bits, conforme indicado no esquema a seguir:

Número de bits	Máscara de sub-rede
08	255.0.0.0
16	255.255.0.0
24	255.255.255.0

Uma máscara de 8 bits significa que todos os bits do primeiro octeto são iguais a 1; uma máscara de 16 bits significa que todos os bits do primeiro e do segundo octeto são iguais a 1 e uma máscara de 24 bits significa que todos os bits dos três primeiros octetos são iguais a 1. Este conceito está ilustrado na tabela a seguir:

Núm. bits	Octeto 01	Octeto 02	Octeto 03	Octeto 04	Máscara
8	11111111	00000000	00000000	00000000	255.0.0.0
16	11111111	11111111	00000000	00000000	255.255.0.0
24	11111111	11111111	11111111	00000000	255.255.255.0

Máscaras de rede com 8, 16 e 24 bits

No exemplo da rede com matriz em São Paulo e mais cinco escritórios, vamos utilizar uma rede classe C, que será subdividida em seis sub-redes (na prática 8, mas a

primeira e a última não são utilizadas). Para fazer esta subdivisão, você deve alterar o número de bits iguais a 1 na máscara de sub-rede. Por exemplo, ao invés de 24 bits, você terá que utilizar 25, 26, 27 ou um número a ser definido. Bem, já avançamos mais um pouco:

"Para fazer a divisão de uma rede em sub-redes, é preciso aumentar o número de bits iguais a 1, alterando com isso a máscara de sub-rede."

Quantos bits devem ser utilizados para a máscara de sub-rede?

Agora, naturalmente, surge uma nova questão: "Quantos bits?". Ou de uma outra maneira (já procurando induzir o seu raciocínio): "O que define o número de bits a ser utilizados a mais?"

Bem, esta é uma questão bem mais simples do que pode parecer. Vamos a ela. No exemplo proposto, precisamos dividir a rede em seis sub-redes. Ou seja, o número de sub-redes deve ser, pelo menos, seis. Sempre lembrando que a primeira e a última sub-rede não são utilizadas. O número de sub-redes é proporcional ao número de bits que vamos adicionar à máscara de sub-rede já existente. O número de rede é dado pela fórmula a seguir, onde 'n' é o número de bits a mais a serem utilizados para a máscara de sub-rede:

1. Núm. de sub-redes = 2n-2

No nosso exemplo estão disponíveis até 8 bits do último octeto para serem também utilizados na máscara de sub-rede. Claro que na prática não podemos usar os 8 bits, senão ficaríamos com o endereço de broadcast: 255.255.255.255, como máscara de sub-rede. Além disso, quanto mais bits pegar para a máscara de sub-rede, menos sobrarão para os números IP da rede. Por exemplo, se adicionar mais um bit a máscara já existente, ficarei com 25 bits para a máscara e 7 para números IP, se adicionar mais dois bits à máscara original de 24 bits, ficarei com 26 bits para a máscara e somente 6 para números IP e assim por diante. O número de bits que restam para os números IP, definem quantos números IP podem haver em cada sub-rede. A fórmula para determinar o número de endereços IP dentro de cada sub-rede, é indicado a seguir, onde 'n' é o númeo de bits destinados a parte de host do

endereço (32 - bits usados para a máscara):

2. Núm. de end. IP dentro de cada sub-rede = 2n-2

Na tabela a seguir, veja os cálculos para a divisão de sub-redes que será feita no exemplo. Observe que quanto mais bits é adicionado à máscara de sub-rede, mais sub-redes é possível obter, porém com um menor número de máquinas em cada sub-rede. Lembrando que no exemplo estamos subdividindo uma rede classe **C - 200.100.100.0/255.255.255.0,** ou seja, uma rede com 24 bits para a máscara de sub-rede original.

Número de bits a mais a serem utilizados	Número de sub-redes	Número de hosts em cada sub-rede
0	máscara original. rede classe C sem divisão	254
1	0	126
2	2	62
3	6	30
4	14	14
5	30	6
6	62	2
7	126	0
8	endereço de broadcast	-

Número de redes e número de hosts em cada rede.

Claro que algumas situações não se aplicam na prática. Por exemplo, usando apenas um bit a mais para a máscara de sub-rede, isto é, 25 bits ao invés de 24. Neste caso teremos 0 sub-redes disponíveis. Pois com 1 bit é possível criar apenas duas sub-redes, como a primeira e a última são descartadas, conforme descrito anteriormente, na prática as duas sub-redes geradas não poderão ser utilizadas. A mesma situação ocorre com o uso de 7 bits a mais para a máscara de sub-rede, ou seja, 31 ao invés de 24. Nesta situação sobra apenas um bit para os endereços IP. Com 1 bit posso ter apenas dois endereços IP, descontando o primeiro e o último que não são utilizados, não sobra nenhum endereço IP. As situações intermediárias é que são mais realistas. No nosso exemplo, precisamos dividir a rede Classe C - 200.100.100.0/255.255.255.0, em seis sub-redes. De acordo com a tabela da Figura anterior, precisamos utilizar 3 bits a mais para obter as seis sub-redes dese-

jadas.

Observe que utilizando três bits a mais, ao invés de 24 bits (máscara original), vamos utilizar 27 bits para a máscara de sub-rede. Com isso sobram cinco bits para os números IPs dentro de cada sub-rede, o que dá um total de 30 números IP por sub-rede. Exatamente o que precisamos.

A próxima questão que pode surgir é como é que fica a máscara de sub-rede, agora que ao invés de 24 bits, estou utilizando 27 bits, conforme ilustrado na tabela a seguir:

Núm. bits	Octeto 01	Octeto 02	Octeto 03	Octeto 04
27	11111111	11111111	11111111	11100000

Figura - Máscara de sub-rede com 27 bits.

Para determinar a nova máscara temos que revisar o valor de cada bit. Da esquerda para a direita, cada bit representa o seguinte valor, respectivamente:

128 64 32 16 8 4 2 1

Como os três primeiros bits do último octeto foram também utilizados para a máscara, estes três bits somam para o valor do último octeto. No nosso exemplo, o último octeto da máscara terá o seguinte valor: 128+64+32 = 224. Com isso a nova máscara de sub-rede, máscara esta que será utilizada pelas seis sub-redes, é a seguinte: 255.255.255.224. Observe que ao adicionar bits à máscara de sub-rede, fazemos isso a partir do bit de maior valor, ou seja, o bit mais da esquerda, com o valor de 128, depois usamos o próximo bit com valor 64 e assim por diante. Na tabela a seguir, apresento a ilustração de como fica a nova máscara de sub-rede:

Núm. bits	Octeto 01	Octeto 02	Octeto 03	Octeto 04	Nova Máscara
27	11111111	11111111	11111111	11100000	255.255.255.224

Figura - Como fica a nova máscara de sub-rede.

Com o uso de três bits adicionais para a máscara de rede, teremos seis sub-redes

disponíveis (uma para cada escritório) com um número máximo de 30 números IP por sub-rede. Exatamente o que precisamos para o exemplo proposto.A idéia básica de subnet é bastante simples. Utiliza-se bits adicionais para a máscara de sub-rede. Com isso se tem uma divisão da rede original (classe A, classe B ou classe C) em várias sub-redes, sendo que o número de endereços IP em cada sub-rede é reduzido (por termos utilizados bits adicionais para a máscara de sub-rede, bits estes que originalmente eram destinados aos endereços IP). Esta divisão pode ser feita em redes de qualquer uma das classes padrão A, B ou C. Por exemplo, por padrão, na Classe A são utilizados 8 bits para a máscara de sub-rede e 24 bits para hosts. Você pode utilizar, por exemplo, 12 bits para a máscara de sub-rede, restando com isso 20 bits para endereços de host.

Na tabela a seguir, é apresentado os cálculos para o número de sub-redes e o número de hosts dentro de cada sub-rede, apenas para os casos que podem ser utilizados na prática, ou seja, duas ou mais sub-redes e dois ou mais endereços válidos em cada sub-rede, quando for feita a sub-divisão de uma rede Classe C, com máscara original igual a 255.255.255.0..

Número de bits a mais a serem utilizados	Número de sub-redes	Número de hosts em cada sub-rede
2	2	62
3	6	30
4	14	14
5	30	6
6	62	2

Número de redes e número de hosts em cada rede - divsão de uma rede Classe C.

Lembrando que a fórmula para calcular o número de sub-redes é:

Núm. de sub-redes = 2n-2

onde n é o número de bits a mais utilizados para a máscara de sub-rede E a fórmula para calcular o número de endereços IP dentro de cada sub-rede é:

Núm de IPs por subrede = 2n-2

onde n é o número de bits restantes, isto é, não utilizados pela máscara de sub-rede.

Até aqui foram vistos ememplos da rede Classe C, que está sendo subdividida em várias sub-redes. Porém é também possível subdividir redes Classe A e redes Classe B. Lembrando que redes classe A utilizam, por padrão, apenas 8 bits para o endereço de rede, já redes classe B, utilizam, por padrão, 16 bits. Na tabela a seguir, apresento um resumo do número de bits utilizados para a máscara de sub-rede, por padrão, nas classes A, B e C:

Classe	Bits	Octeto 01	Octeto 02	Octeto 03	Octeto 04	Máscara padrão
A	8	11111111	00000000	00000000	00000000	255.0.0.0
B	16	11111111	11111111	00000000	00000000	255.255.0.0
C	24	11111111	11111111	11111111	00000000	255.255.255.0

Figura - Máscara padrão para as classes A, B e C

Para subdividir uma rede classe A em sub-redes, basta usar bits adicionais para a máscara de sub-rede. Por padrão são utilizados 8 bits. Se você utilizar 10, 12 ou mais bits, estará criando sub-redes. O mesmo raciocínio é válido para as redes classe B, as quais utilizam, por padrão, 16 bits para a máscara de sub-rede. Se você utilizar 18, 20 ou mais bits para a máscara de sub-rede, estará subdividindo a rede classe B em várias sub-redes.

As fórmulas para cálculo do número de sub-redes e do número de hosts em cada sub-rede são as mesmas apresentadas anteriormente, independentemente da classe da rede que está sendo dividida em sub-redes.

A seguir é apresentado uma tabela com o número de sub-redes e o número de hosts em cada sub-rede, dependendo do número de bits adicionais (além do padrão definido para a classe) utilizados para a máscara de sub-rede, para a divisão de uma rede Classe B:

Divisão de uma rede classe B em sub-redes			
Número de bits	Sub-redes	Hosts	Nova máscara de sub-rede
2	2	16382	255.255.192.0
3	6	8190	255.255.224.0
4	14	4094	255.255.240.0
5	30	2046	255.255.248.0
6	62	1022	255.255.252.0
7	126	510	255.255.254.0
8	254	254	255.255.255.0
9	510	126	255.255.255.128
10	1022	62	255.255.255.192
11	2046	30	255.255.255.224
12	4094	14	255.255.255.240
13	8190	6	255.255.255.248

Tabela - Número de redes e número de hosts em cada rede - Classe B.

Observe como o entendimento dos cálculos binários realizados pelo TCP/IP facilita o entendimento de vários assuntos relacionados ao TCP/IP, inclusive o conceito de subnet. Por padrão a classe B utiliza 16 bits para a máscara de sub-rede, ou seja, uma máscara padrão: 255.255..0.0. Agora se utilizar oito bits adicionais (todo o terceiro octeto) para a máscara, terá todos os bits do terceiro octeto como sendo iguais a 1, com isso a máscara passa a ser: 255.255.255.0. Este resultado está coerente com a tabela da Figura 16.11. Agora ao invés de 8 bits adicionais, utilize 9. Ou seja, todo o terceiro octeto (8 bits) mais o primeiro bit do quarto octeto. O primeiro bit, o bit bem à esquerda é o bit de valor mais alto, ou seja, o que vale 128. Ao usar este bit também para a máscara de sub-rede, será obtida a seguinte máscara: 255.255.255.128. Também fecha com a tabela anterior. Com isso se conclui que o entendimento da aritemética e da representação binária, facilita muito o estudo do protocolo TCP/IP e de assuntos relacionados, tais como subnet e roteamento.

A seguir é apresentada uma tabela com o número de sub-redes e o número de hosts em cada sub-rede, dependendo do número de bits adicionais (além do padrão definido para a (classe) utilizados para a máscara de sub-rede, para a divisão de uma rede Classe A:

Divisão de uma rede classe C em sub-redes			
Número de bits	Sub-redes	Hosts	Nova máscara de sub-rede
2	2	4194302	255.192.0.0
3	6	2097150	255.224.0.0
4	14	1048574	255.240.0.0
5	30	524286	255.248.0.0
6	62	262142	255.252.0.0
7	126	131070	255.254.0.0
8	254	65534	255.255.0.0
9	510	32766	255.255.128.0
10	1022	16382	255.255.192.0
11	2046	8190	255.255.224.0
12	4094	4094	255.255.240.0
13	8190	2046	255.255.248.0
14	16382	1022	255.255.252.0
15	32766	510	255.255.254.0
16	65534	254	255.255.255.0
17	131070	126	255.255.255.128
18	262142	62	255.255.255.192
19	524286	30	255.255.255.224
20	1048574	14	255.255.255.240
21	2097150	6	255.255.255.248
22	4194302	2	255.255.255.252

Tabela - Número de redes e número de hosts em cada rede - Classe A.

Um fato importante, que é destacado novamente é que todas as sub-redes (resultantes da divisão de uma rede), utilizam o mesmo número para a máscara de sub-rede. Por exemplo, na quarta linha da tabela indicada na Figura 16.12, é utilizado 5 bits adicionais para a máscara de sub-rede, o que resulta em 30 sub-redes diferentes, porém todas utilizando como máscara de sub-rede o seguinte número: 255.248.0.0.

Muito bem, entendido o conceito de divisão em sub-redes e de determinação do número de sub-redes, do número de hosts em cada sub-rede e de como é formada a nova máscara de sub-rede, a próxima questão que pode surgir é a seguinte:

Como listar as faixas de endereços para cada sub-rede? Este é exatamente o assunto que vem a seguir.

Como listar as faixas de endereços dentro de cada sub-rede

Vamos entender esta questão através de exemplos práticos.

Exemplo 01: Dividir a seguinte rede classe C: 229.45.32.0/255.255.255.0. São necessárias, pelo menos, 10 sub-redes. Determinar o seguinte:

a) Quantos bits serão necessários para fazer a divisão e obter pelo menos 10 sub-redes?

b) Quantos números IP (hosts) estarão disponíveis em cada sub-rede?

c) Qual a nova máscara de sub-rede?

d) Listar a faixa de endereços de cada sub-rede. Vamos ao trabalho. Para responder a questão da letra a, você deve lembrar da fórmula:

Núm. de sub-redes = 2n-2

Você pode ir substituindo n por valores sucessivos, até atingir ou superar o valor de 10. Por exemplo, para n=2, a fórmula resulta em 2, para n=3, a fórmula resulta em 6, para n=4 a fórmula resulta em 14. Bem, está respondida a questão da letra a, temos que utilizar quatro bits do quarto octeto para fazer parte da máscara de sub-rede.

a) Quantos bits serão necessários para fazer a divisão e obter pelo menos 10 sub-redes? **R:** 4 bits.

Como utilizei quatro bits do último octeto (além dos 24 bits dos três primeiros octetos, os quais já faziam parte da máscara original), sobraram apenas 4 bits para os endereços IP, ou seja, para os endereços de hosts em cada sub-rede. Tenho que lembrar da seguinte fórmula: Núm. de end. IP dentro de cada sub-rede = 2n-2 substituindo n por 4, vou obter um valor de 14. Com isso já estou em condições de responder a alternativa b.

b) Quantos números IP (hosts) estarão disponíveis em cada sub-rede? **R:** 14.

Como utilizei quatro bits do quarto octeto para fazer a divisão em sub-redes, os quatro primeiros bits foram definidos iguais a 1. Basta somar os respectivos valores, ou seja: 128+64+32+16 = 240. Ou seja, com os quatro primeiros bits do quarto octeto sendo iguais a 1, o valor do quarto octeto passa para 240, com isso já temos condições de responder a alternativa c.

c) Qual a nova máscara de sub-rede? **R:** 255.255.255.240

É importante lembrar, mais uma vez, que esta será a máscara de sub-rede utilizada por todas as 14 sub-redes.

d) Listar a faixa de endereços de cada sub-rede. Esta é a novidade deste item. Como saber de que número até que número vai cada endereço IP. Esta também é fácil, embora seja novidade. Observe o último bit definido para a máscara. No nosso exemplo é o quarto bit do quarto octeto. Qual o valor decimal do quarto bit? 16 (o primeiro é 128, o segundo 64, o terceiro 32 e assim por diante, conforme explicado na Parte 2). O valor do último bit é um indicativo das faixas de variação para este exemplo. Ou seja, na prática temos 16 hosts em cada sub-rede, embora o primeiro e o último não devam ser utilizados, pois o primeiro é o endereço da própria sub-rede e o último é o endereço de broadcast da sub-rede. Por isso que ficam 14 hosts por sub-rede, devido ao -2"na fórmula, o -2"significa: - o primeiro - o último. Ao listar as faixas, consideramos os 16 hosts, apenas é importante salienar que o primeiro e o último não são utilizados. Com isso a primeira sub-rede vai do host 0 até o 15, a segunda sub-rede do 16 até o 31, a terceira do 32 até o 47 e assim por diante, conforme indicado no esquema a seguir:

Divisão da rede em 14 sub-redes, onde cada sub-rede fica com 16 endereços IP, sendo que a primeira e a última sub-rede não são utilizadas e o primeiro e o último número IP, dentro de cada sub-rede, também não são utilizados:

Sub-rede 01 229.45.32.0 -> 229.45.32.15

Sub-rede 02 229.45.32.16 -> 229.45.32.31

Sub-rede 03 229.45.32.32 -> 229.45.32.47

Sub-rede 04 229.45.32.48 -> 229.45.32.63

Sub-rede 05 229.45.32.64 -> 229.45.32.79

Sub-rede 06 229.45.32.80 -> 229.45.32.95

Sub-rede 07 229.45.32.96 -> 229.45.32.111

Sub-rede 08 229.45.32.112 -> 229.45.32.127

Sub-rede 09 229.45.32.128 -> 229.45.32.143

Sub-rede 10 229.45.32.144 -> 229.45.32.159

Sub-rede 11 229.45.32.160 -> 229.45.32.175

Sub-rede 12 229.45.32.176 -> 229.45.32.191

Sub-rede 13 229.45.32.192 -> 229.45.32.207

Sub-rede 14 229.45.32.208 -> 229.45.32.223

Sub-rede 15 229.45.32.224 -> 229.45.32.239

Sub-rede 16 229.45.32.240 -> 229.45.32.255

Vamos a mais um exemplo prático, agora usando uma rede classe B, que tem inicialmente, uma máscara de sub-rede: 255.255.0.0

Exemplo 02: Dividir a seguinte rede classe B: 150.100.0.0/255.255.0.0. São necessárias, pelo menos, 20 sub-redes. Determinar o seguinte:

a) Quantos bits serão necessários para fazer a divisão e obter pelo menos 10 sub-redes?

b) Quantos números IP (hosts) estarão disponíveis em cada sub-rede?

c) Qual a nova máscara de sub-rede?

d) Listar a faixa de endereços de cada sub-rede. Vamos ao trabalho. Para responder a questão da letra a, você deve lembrar da fórmula:

Núm. de sub-redes = 2n-2

Você pode ir substituindo n por valores sucessivos, até atingir ou superar o valor de 10. Por exemplo, para n=2, a fórmula resulta em 2, para n=3, a fórmula resulta em 6, para n=4 a fórmula resulta em 14 e para n=5 a fórmula resulta em 30. Bem, está respondida a questão da letra a, temos que utilizar cinco bits do terceiro octeto para fazer parte da máscara de sub-rede. Pois se utilizar apenas 4 bits, obterá somente 14 sub-redes e usando mais de 5 bits, obterá um número de sub-redes bem maior do que o necessário.

a) Quantos bits serão necessários para fazer a divisão e obter pelo menos 20 sub-redes?

R: 5 bits.

Como utilizei cinco bits do terceiro octeto (além dos 16 bits dos dois primeiros octetos, os quais já faziam parte da máscara original), sobraram apenas 11 bits (os três restantes do terceiro octeto mais os 8 bits do quarto octeto) para os endereços IP, ou seja, para os endereços de hosts em cada sub-rede. Lembre-se da seguinte fórmula:

Núm. de endereços IP dentro de cada sub-rede = 2n-2 Substituindo n por 11 (número de bits que restarama para a parte de host), vou obter um valor de 2046, já descontando o primeiro e o último número, os quais não podem ser utilizados, conforme já descrito anteriormente. Com isso já estou em condições de responder a alternativa b.

b) Quantos números IP (hosts) estarão disponíveis em cada sub-rede?

R: 2046.

Como utilizei cinco bits do terceiro octeto para fazer a divisão em sub-redes, os cinco primeiros bits foram definidos iguais a 1. Basta somar os respectivos valores, ou seja: 128+64+32+16+8 = 248. Ou seja, com os quatro primeiros bits do quarto octeto sendo iguais a 1, o valor do quarto octeto passa para 248, com isso já temos

condições de responder a alternativa c.

c) Qual a nova máscara de sub-rede?

R: 255.255.248.0 É importante lembrar, mais uma vez, que esta será a máscara de sub-rede utilizada por todas as 30 sub-redes.

d) Listar a faixa de endereços de cada sub-rede. Como saber de que número até que número vai cada endereço IP. Esta também é fácil e o raciocínio é o mesmo utilizado para o exemplo anterior, onde foi feita uma divisão de uma rede classe C. Observe o último bit definido para a máscara. No nosso exemplo é o quinto bit do terceiro octeto. Qual o valor decimal do quinto bit (de qualque octeto)? 8 (o primeiro é 128, o segundo 64, o terceiro 32, o quarto é 16 e o quinto é 8. O valor do último bit é um indicativo das faixas de variação para este exemplo. Ou seja, na prática temos 2048 hosts em cada sub-rede, embora o primeiro e o último não devam ser utilizados, pois o primeiro é o endereço da própria sub-rede e o último é o endereço de broadcast da sub-rede. Por isso que ficam 2046 hosts por sub-rede, devido ao -2"na fórmula, o -2"significa: - o primeiro - o último. Ao listar as faixas, consideramos o valor do último bit da máscara. No nosso exemplo é o 8. A primeira faixa vai do zero até um número anterior ao valor do último bit, no caso do 0 ao 7. A seguir indico a faixa de endereços da primeira sub-rede (sub-rede que não será utilizada na prática, pois descarta-se a primeira e a última):

Sub-rede 01 150.100.0.1 -> 150.100.7.254

Com isso todo endereço IP que tiver o terceiro número na faixa entre 0 e 7, será um número IP da primeira sub-rede, conforme os exemplos a seguir:

150.100.0.25

150.100.3.20

150.100.5.0

150.100.6.244

Importante: Observe que os valores de 0 a 7 são definidos no terceiro octeto, que é onde estamos utilizando cinco bits a mais para fazer a divisão em sub-redes.

Qual seria a faixa de endereços IP da próxima sub-rede. Aqui vale o mesmo reciocínio. O último bit da máscara equivale ao valor 8. Esta é a variação da terceira parte do número IP, que é onde esta sendo feita a divisão em sub-redes. Então, se a primeira foi de 0 até 7, a segunda sub-rede terá valores de 8 a 15 no terceiro octeto, a terceira sub-rede terá valores de 16 a 23 e assim por diante.

Divisão da rede em 32 sub-redes, onde cada sub-rede fica com 2048 endereços IP, sendo que a primeira e a última sub-rede não são utilizadas e o primeiro e o último número IP, dentro de cada sub-rede, também não são utilizados:

Sub-rede	Primeiro IP	Último IP	End. de broadcast	Número
150.100.0.0	150.100.0.1	150.100.7.254	150.100.7.255	01
150.100.8.0	150.100.8.1	150.100.15.254	150.100.15.255	02
150.100.16.0	150.100.16.1	150.100.23.254	150.100.23.255	03
150.100.24.0	150.100.24.1	150.100.31.254	150.100.31.255	04
150.100.32.0	150.100.32.1	150.100.39.254	150.100.39.255	05
150.100.40.0	150.100.40.1	150.100.47.254	150.100.47.255	06
150.100.48.0	150.100.48.1	150.100.55.254	150.100.55.255	07
150.100.56.0	150.100.56.1	150.100.63.254	150.100.63.255	08
150.100.64.0	150.100.64.1	150.100.71.254	150.100.71.255	09
150.100.72.0	150.100.72.1	150.100.79.254	150.100.79.255	10
150.100.80.0	150.100.80.1	150.100.87.254	150.100.87.255	11
150.100.88.0	150.100.88.1	150.100.95.254	150.100.95.255	12
150.100.96.0	150.100.96.1	150.100.103.254	150.100.103.255	13
150.100.104.0	150.100.104.1	150.100.111.254	150.100.111.255	14
150.100.104.0	150.100.104.1	150.100.111.254	150.100.111.255	14
150.100.112.0	150.100.112.1	150.100.119.254	150.100.119.255	15
150.100.120.0	150.100.120.1	150.100.127.254	150.100.127.255	16
150.100.128.0	150.100.128.1	150.100.135.254	150.100.135.255	17
150.100.136.0	150.100.136.1	150.100.143.254	150.100.143.255	18
150.100.144.0	150.100.144.1	150.100.151.254	150.100.151.255	19
150.100.152.0	150.100.152.1	150.100.159.254	150.100.159.25520	

150.100.160.0	150.100.160.1	150.100.167.254	150.100.167.255	21
150.100.168.0	150.100.168.1	150.100.175.254	150.100.175.255	22
150.100.176.0	150.100.176.1	150.100.183.254	150.100.183.255	23
150.100.184.0	150.100.184.1	150.100.191.254	150.100.191.255	24
150.100.192.0	150.100.192.1	150.100.199.254	150.100.199.255	25
150.100.200.0	150.100.200.1	150.100.207.254	150.100.207.255	26
150.100.208.0	150.100.208.1	150.100.215.254	150.100.215.255	27
150.100.216.0	150.100.216.1	150.100.223.254	150.100.223.255	28
150.100.224.0	150.100.224.1	150.100.231.254	150.100.231.255	29
150.100.232.0	150.100.232.1	150.100.239.254	150.100.239.255	30
150.100.240.0	150.100.240.1	150.100.247.254	150.100.247.255	31
150.100.248.0	150.100.2481	150.100.255.254	150.100.255.255	32

Com base na tabela apresentada, fica fácil responder em que sub-rede está contido um determinado número IP. Por exemplo, considere o número IP 150.100.130.222. Primeiro você observa o terceiro octeto do número IP (o terceiro, porque é neste octeto que estão os últimos bits que foram utilizados para a máscara de sub-rede). Consultando a tabela anterior, você observa o valor de 130 para o terceiro octeto corresponde a sub-rede 17, na qual o terceiro octeto varia entre 128 e 135, conforme indicado a seguir:

11.2.2 Ipv6

"IPv6 é a nova versão do protocolo de redes de dados nos quais a Internet está baseada. O IETF (Internet Engineering Task Force), desenvolveu suas especificações básicas durante os anos 90. A principal motivação para o desenvolvimento e lançamento do IPv6 foi a expansão do espaço de endereços disponíveis na Internet, permitindo assim que se conectem bilhões de novos dispositivos (PDAs, telefones celulares, etc), novos usários e tecnologias sempre-conectada (xDSL, cabo, Ethernet ou fibra direto na residência, comunicação via rede elétrica, etc).

Não existem classes como A, B e C. O IPv6 utiliza o conceito de CIDR (FULLER, 1993), onde um determinado número de bits corresponde ao prefixo da rede, e os bits restantes identificam o nó.

O protocolo ARP não é utilizado em IPv6, pois não existe broadcast em IPv6 e o ARP baseia-se em broadcast. Em seu lugar, é utilizado o protocolo ICMPv6 e transmissão *multicast.*

Espaço de endereçamento

O espaço de endereçamento do IPv6 é de 128 bits, contra os 32 bits do IPv4. Esta é a mudança mais visível do IPv6 em relação ao IPv4. Algumas das primeiras propostas de evolução do IPv4 - vide CALLON (1992), PISTICELLO (1993) e BRADNER & MANKIN (1993) - propunham espaços de endereçamento de 64 ou 96 bits, perfeitamente suficientes para um prazo razoavelmente longo.

A proposta mais interessante, denominada TUBA (TCP and UDP with Bigger Addresses) propunha a substituição do IP pelo CNLP da pilha OSI. O CNLP é bem documentado e tem um espaço de endereçamento de até 20 octetos (160 bits). A indisposição generalizada da comunidade Internet com o protocolo OSI, constatada no trabalho de DIXON (1993), acabou sepultando a idéia. Textos a favor e contra o TUBA e o OSI podem ser encontrados facilmente na Internet.

O endereçamento finalmente adotado visa, principalmente: a) abrir espaço à criação de tantas classes de endereços quantas forem necessárias, e ainda ter espaço de sobra para um número virtualmente inesgotável de endereços dentro de cada classe;

b) utilização massiva de roteamento por agregação, onde todas as sub-redes de uma mesma rede apresentam o mesmo prefixo de rede. Isto diminui drasticamente o número de rotas que cada roteador tem de conhecer, em todos os níveis.

Embora o roteamento por agregação seja padrão para IPv4 desde 1995 com a implementação da CIDR (FULLER, 1993), nem todas as redes classe A, B ou C podem ser renumeradas, e os roteadores da espinha dorsal da Internet têm de conhecer rotas específicas para inúmeras redes não agregadas.

O tamanho do endereço IPv6 comporta tanto profundas hierarquias de endereçamento por agregação bem como um grande número de nós por sub-rede. Isso per-

mite:

a) liberal distribuição de faixas de endereçamento a usuários finais, tornando desnecessários, por exemplo, os complexos roteadores NAT (Network Address Translation – tradução de endereço de rede) para compartilhamento de um IP por vários usuários. O IPv6 acaba com os cidadãos de segunda classe da Internet;

b) Com o desuso do NAT, ocorre uma grande simplificação na configuração de servidores e dispositivos de rede, o que contribui para o barateamento do acesso à Internet. Evita todos os problemas citados por PEÑA (2001) e permite que apareçam protocolos mais sofisticados como voz sobre IP.

Nada impede de um sistema operacional ou dispositivo de rede implementar NAT para IPv6, e de fato é implementado no Linux. Alguns administradores de rede têm a sensação subjetiva de que NAT aumenta a segurança, embora isso seja muito discutível.

11.3 Tipos de Endereços IPv6

O grande espaço de endereçamento visa a criação facilitada de classes de endereçamento. Tais classes, mais apropriadamente denominadas de faixas de endereçamento, são registradas junto à IETF. Segue uma lista das principais faixas e os respectivos prefixos IPv6.

0000::/8	Reservado
0000::/96	Endereços IPv6 compatíveis com IPv4
::FFFF:0:0/96	Endereços IPv4 mapeados em IPv6
0200::/8	NSAP (obsoleto)
0400::/8	IPX (obsoleto)
2000::/3	Endereços roteáveis na Internet (prefixos **2xxx** e **3xxx**)
FE80::/10	Endereços da rede local (automáticos, estáticos ou *stateless*)
FEC0::/19	Endereços do sítio local
FF00::/8	*Multicast*

Aproximadamente 15% do espaço de endereçamento IPv6 foi alocado. Restam ainda 85%.

Segundo a RFC 2374, uma mesma interface, que utiliza o protocolo IPv6, pode utilizar mais de um endereço, diferentemente do IPv4, onde tal característica só era possível em roteadores. Essa característica é importante porque na versão 6 algumas aplicações, em geral de controle, utilizam-se de endereços especiais que veremos adiante. Para o endereçamento das interfaces existem então 3 tipos de endereços:

- Unicast;
- Anycast;
- Multicast.

Outra característica marcante do IPv6 é que não existem mais os endereços broadcast, que endereçavam todos os hosts de um mesmo domínio de colisão, isto é, uma pacote com endereço de destino do tipo broadcast era enviado para todos os hosts de seu domínio de colisão. Com a abolição desse tipo endereço, outro protocolo muito comum no IPv4 também ficou em desuso, o ARP – Address Resolution Protocol, que usava endereços broadcast para descoberta do endereço MAC da interface referente ao endereço de destino do pacote.

11.3.1 Endereços Unicast

Esse tipo de endereço é comumente usado em IPv4, que identifica apenas uma única interface. Desta forma um pacote destinado a um endereço do tipo Unicast é enviado diretamente para a interface associada a esse endereço. Foram definidos pela RFC 2374 vários tipos de endereços Unicast :

Agregatable Global Unicast Address

- Loopback Address

- Unspecified Address

- NSAP Address

- IPX AddressSite-local Unicast Address

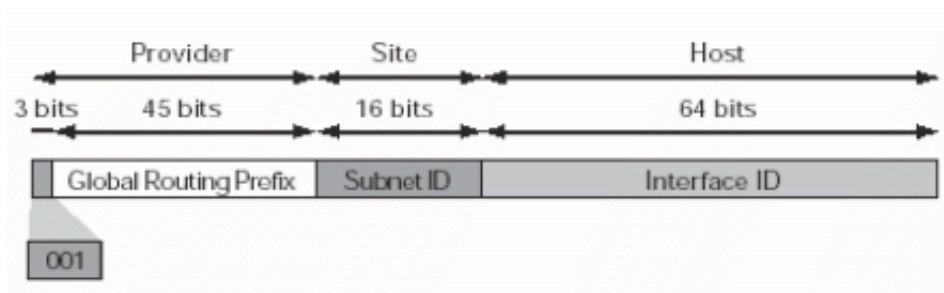
- Link-local Unicast Address

- IPv4-compatible IPv6 Address

Agregatable Global Unicast Address

Esse tipo de endereço unicast é equivalente ao endereço global unicast usado em IPv4. Sendo assim é o endereço que será usado globalmente na Internet. Essa estrutura de endereços globais permite uma agregação de prefixos de roteamento que limitam o número de entradas nas tabelas de rotas.

A estrutura deste tipo de endereço é dividida em 4 níveis, o primeiro é o FP - Format Prefix, que indica justamente que se trata de um endereço do tipo Global Unicast, esse FP deve ser sempre 001, como vimos na tabela III - Alocação de endereços IPv6, na seção anterior.

O segundo campo é chamado Global Routing Prefix, e é destinado a identificação dos ISP´s - Internet Service Provider, ele possui vários níveis e seguem a estrutura apresentada na seção anterior. O terceiro campo Subnet ID também foi apresentado anteriormente como sendo o campo Site ID da estrutura de hierarquização do endereço IPv6, o último nível é o Interface ID, que também já foi abordado e possui 64 bits. Abaixo, vemos na figura a estrutura desse tipo de endereço:

Estrutura do endereço Aggregatable Global Unicast Address.

Loopback Address

Esse tipo de endereço, como o próprio nome já diz, é o endereço da própria interface. Porém ele só pode ser usado quando um nó envia um pacote para ele mesmo. No IPv4 esse tipo de endereço era geralmente o 127.0.0.1, em IPv6 é indicado por: **0:0:0:0:0:0:0:1**

ou simplesmente:

::1

Esse endereço não pode ser associado a nenhuma interface física, nem como endereço de fonte, nem como endereço de destino, mas pode ser imaginado como sendo de uma interface virtual, a interface loopback. Um pacote IPv6 com endereço destino do tipo loopback address também não deve deixar o próprio host, sendo que esse endereço nunca será repassado por um roteador IPv6.

Unspecified Address

Esse tipo de endereço indica exatamente a ausência de um endereço. Ele nunca deverá ser utilizado como um endereço válido para nenhum host. A sua utilidade é para que estações que ainda não foram inicializadas, sejam identificadas com endereços deste tipo, ou seja, hosts que ainda não tenham aprendido seus próprios endereços globais, utilizem tais endereços para se autoconfigurar. Além disso, esse tipo de endereço não deve ser utilizado como endereço de destino ou em cabeçalho

de roteamento de pacotes IPv6. Seu formato é o seguinte:

0:0:0:0:0:0:0:0

ou simplesmente:

::

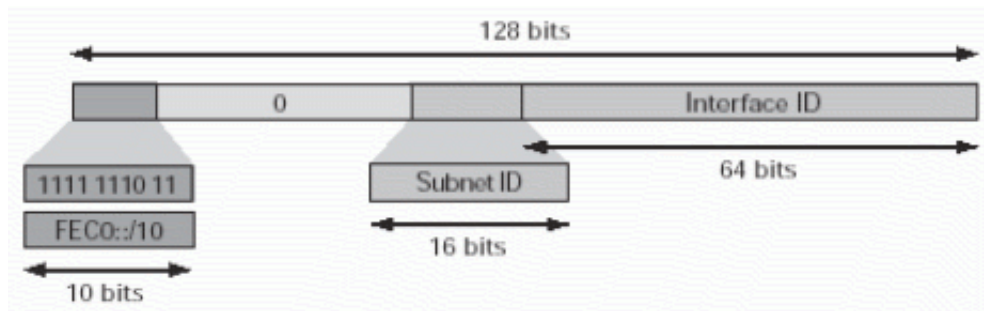
NSAP Address

Esse tipo de endereço é ident ificado pelo prefixo FP - 0000001. Ele foi definido pela RFC 1888 - OSI NSAPs and IPv6 como mecanismo de suporte para endereçamento OSI NSAP - Network Service Access Point em redes IPv6. Possui além do FP de 7 bits, que o indica, 121 bits para constituição de seu endereço. IPX Address

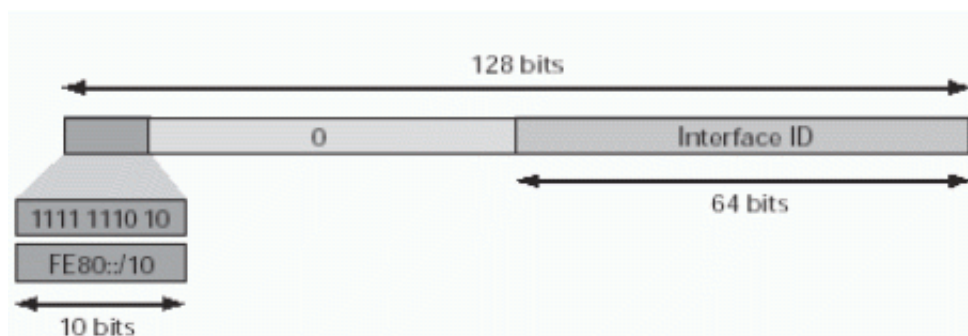
É também um endereço especial para compatibilidade de endereços existentes. É identificado pelo prefixo 0000010, incluído para prover mecanismo de mapeamento de endereços IPX - Internal Packet eXchange em endereços IPv6. Os endereços IPX são utilizados em redes Netware, de propriedade da Novell. Da mesma forma que o NSAP Address possui 7 bits de FP e 121 bits para constituição do endereço.

Site Local Unicast Address

O endereço do tipo Site Local é similar aos endereços privados usados em IPv4, como as redes 10.0.0.0 /8, 172.16.0.0/16 e 198.168.0.0/16. Esses endereços podem ser usados para uma comunicação restrita dentro de um domínio específico.

Este tipo de endereço é identificado pelo prefixo **FEC0::/10 ou 1111111011** em binário. Ele pode ser definido para uso interno numa organização através da concatenação do campo de SLA (16 bits) com a identificação da interface (64 bits). Este tipo de endereçamento pode ser considerado como privado, visto que ele está restrito a um domínio sem ligação à Internet. Desta forma ele não pode ser anunciado externamente por roteadores. Abaixo podemos visualizar a estrutura deste tipo de endereço.

Estrutura do endereço Site Local Unicast Address.

Link Local Unicast Address

Este tipo de endereço é automaticamente configurado em qualquer host IPv6, através da conjugação do seu prefixo **FE80::/10 ou 1111111010** em binário, como pode ser visto na tabela III, e a identificação da interface no formato EUI-64, mostrado anteriormente. Estes endereços são utilizados nos processos de configuração dinâmica automática (autoconfiguração) e no processo de descoberta de elementos na hierarquia de roteamento (Neighbor Discovery Protocol). Estes procedimentos serão vistos com detalhes na próxima seção. Este endereçamento permite também a comunicação entre nós pertencentes ao mesmo enlace. Como nos endereços Site Local, esse tipo de endereço não deve ser enviado como endereço de origem ou destino em pacotes. Além disso esses endereços não são repassados pelos roteadores. Abaixo podemos visualizar a estrutura deste tipo de endereço.

Estrutura do endereço Site Local Unicast Address.

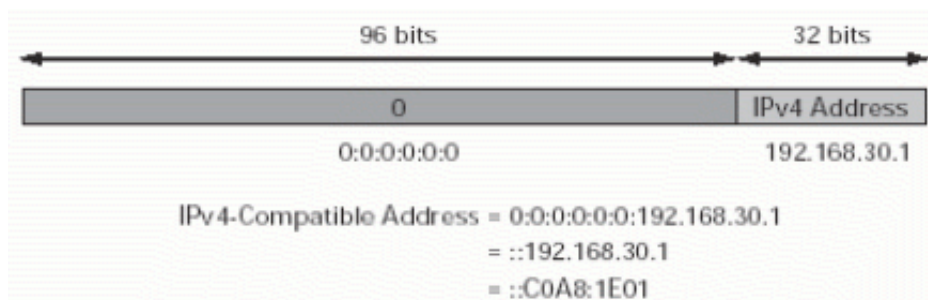
IPv4-compatible IPv6 Address

Esse tipo de endereço é usado em IPv6 como um mecanismo de transição entre IPv6 e IPv4. É utilizado como endereços de destino e origem em tunnel (encapsulamento de um protocolo sobre outro) IPv6 sobre IPv4. É representado por um endereço IPv6 cujos últimos 32 bits são um endereço IPv4. Desta forma, anexando-se um prefixo nulo (96 bits de zeros) a um endereço IPv4 (32 bits) obtém-se o seguinte formato:

0:0:0:0:0:0:192.168.30.1

ou no seu formato abreviado

::192.168.30.1

Abaixo é mostrada a estrutura deste endereço.

Estrutura do endereço IPv6 compatible IPv4 Address.

11.4 Endereços Anycast

Esse tipo de endereço é utilizado para identificar um grupo de interfaces pertencentes a hosts diferentes. Um pacote destinado a um endereço Anycast é enviado para um das interfaces identificadas pelo endereço. Especificamente, o pacote é enviado para a interface mais próxima, de acordo com o protocolo de roteamento.

Um endereço do tipo Anycast não pode ser utilizado como endereço de origem de um pacote IPv6. Este tipo de endereçamento será útil na detecção rápida de um

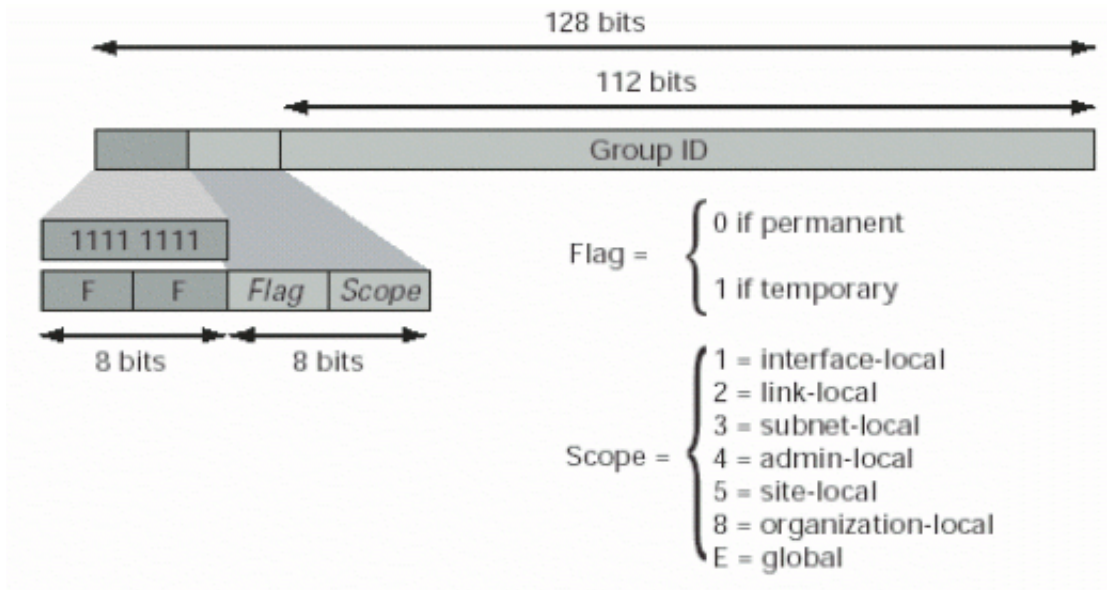
determinado servidor ou serviço. Por exemplo, poderá ser definido um grupo de servidores de DNS configurados com endereçamento Anycast, assim um host irá alcançar o servidor mais próximo utilizando este tipo de endereço.

Existe um prefixo mais longo desse mesmo endereço para cada endereço Anycast atribuído que identifica a região ao qual todas as interfaces pertencem. Abaixo é mostrada a estrutura básica deste tipo de endereço.

Estrutura do endereço Anycast.

11.4.1 Endereço Multicast

Da mesma forma que o endereço Anycast, este endereço identifica um grupo de interfaces pertencente a diferentes hosts mas um pacote destinado a um endereço Multicast é enviado para todas as interfaces que fazem parte deste grupo.

Um endereço do tipo Multicast Address é um endereço IPv6, que é indicado pelo prefixo FP, como visto na tabela III, **FF00::/8 ou 11111111** em binário. O segundo octeto que se segue ao prefixo (FP = FF) define o tempo de vida (lifetime), os 4 primeiros bits e o escopo do endereço Multicast, os últimos 4 bits deste octeto. Um endereço com lifetime permanente tem um parâmetro de tempo de vida igual a "0", enquanto um endereço temporário tem o mesmo parâmetro igual a "1". O escopo para este tipo de endereço apresenta os valores já definidos de 1, 2, 3, 4, 5, 8 e "E"(os outros estão reservados para o futuro, sendo que o escopo F já está reservado para ser usado para âmbito galáctico) para identificar um host, enlace, site, organização ou um escopo global, respectivamente. Os outros 112 bits são utilizados para identificar o grupo Multicast. Abaixo, visualizamos a estrutura de um endereço.

Estrutura do endereço Anycast.

Dentro dos endereços Multicast já reservados, podemos identificar alguns endereços especiais utilizados para funções específicas (todos de lifetime permanente):

- **FF01::1 –** Indica todas as interfaces de escopo local, isto é, somente as interfaces de um mesmo host.

- **FF02::1 –** Indica todas as interfaces de um escopo de enlace local, isto é, todos os hosts de um mesmo domínio de colisão.

- **FF01::2 –** Indica todos os roteadores dentro de um escopo local, isto é, todas as interfaces de um mesmo roteador.

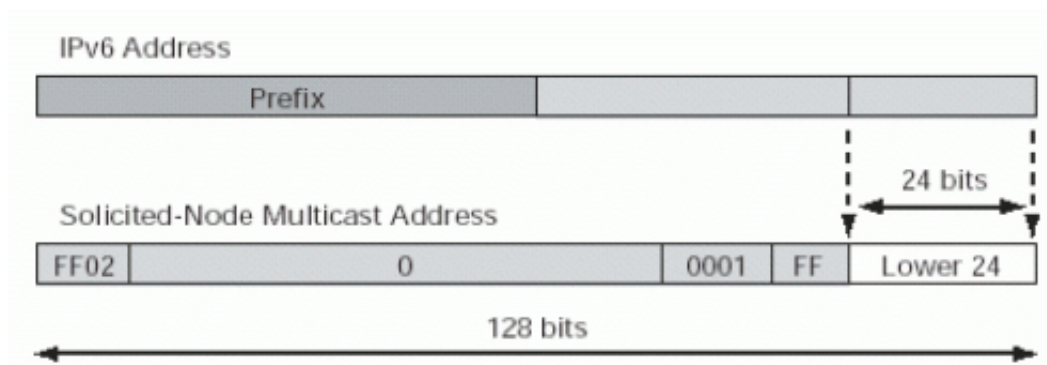
- **FF02::2 –** Indica todos os roteadores dentro de um escopo de enlace local, isto é, todos os roteadores interligados por um mesmo enlace.

- **FF05::2 –** Indica todos os roteadores dentro de um escopo site local, isto é, todos os roteadores que possuem um mesmo site ID.

- **FF02::1:FFxx:xxxx –** Endereço especial chamado de Solicited-Node Multicast

Address, onde xx:xxxx representam os últimos 24 bits do endereço IPv6 Unicast do host.

Solicited-Node Multicast Address

Esse tipo de endereço Multicast especial é usado para mensagens de solicitação de vizinho que auxilia o Neighbor Discovery Protocol e que será visto com mais detalhes na próxima seção. Esse endereço é um grupo Multicast que corresponde a um endereço IPv6 Unicast. A figura abaixo apresenta a estrutura desse endereço.

11.5 Estrutura do endereço Anycast

Comandos avançados de redes:

Configure uma rede para que as máquinas Debian e Red Hat fiquem em redes distintas: rede 192.168.200.0

Com a máscara 255.255.255.240, teremos dezesseis endereços por sub-rede, sendo quatorze utilizáveis, pois o primeiro é reservado para rede e o último pra broadcast. subrede.

Ranges: 192.168.200.

0-15 64-79 128-143 192-207

16-31 80-95 144-159 208-223

32-47 96-111 160-175 224-239

48-63 112-127 176-191 240-255

Para as máquinas Debian vamos utilizar o range 80-95

```
1 # ifconfig eth0 192.168.200.81 netmask 255.255.255.255.248
```

Para as máquinas CentOS vamos utilizar o range 240-255:

```
1 # ifconfig eth0 192.168.200.241 netmask 255.255.255.255.240
```

Agora na Debian tente pingar a máquina CentOS e vice-versa:

```
1 # ping 192.168.200.241
```

Não é possível, pois as máquinas estão em sub-redes diferentes.

Agora no cliente CentOS adicione duas placas de rede uma pra cada subrede.

```
1 # ifconfig eth1 192.168.200.82 netmask 255.255.255.255.240
2 # ifconfig eth2 192.168.200.242 netmask 255.255.255.255.240
```

Pingue da máquina cliente CentOS os dois servidores:

```
1 # ping 192.168.200.241
2 # ping 192.168.200.81
```

Agora na Debian tente pingar a máquina CentOS e vice-versa:

```
1 # ping 192.168.200.241
```

Ainda não é possível.

Para que as duas máquina possam se pingar, adicione o cliente Debian como rota:

No Server Debian:

```
1 # route add default gw 192.168.200.82
```

No Server CentOS:

```
1 # route add default gw 192.168.200.242
```

Agora na Debian tente pingar a máquina CentOS e vice-versa:

```
1 # ping 192.168.200.241
```

Novamente no server Debian tente pingar a máquina servidora CentOS e vice-versa:

Mesmo após adicionarmos a rota para o gateway das redes ainda não é possível pingá-los.

Isto ocorre porque o linux por padrão não encaminha pacotes, na máquina gateway (cliente CentOS) ajuste o encaminhamento:

Visualize o bloqueio do encaminhamento de pacotes:

```
# cat /proc/sys/net/ipv4/ip_forward
```

1 = habilitado 0 = desabilitado

Altere seu valor temporariamente:

```
# echo 1 > /proc/sys/net/ipv4/ip_forward
```

Para ficar permanente edite o arquivo /etc/sysctl.conf e adicione a linha abaixo:

```
# vim /etc/sysctl.conf
net.ipv4.ip_forward = 1
```

Salve o arquivo e execute o comando seguinte para validar a regra e ficar permanente:

```
# sysctl -p
```

11.6 ARP - Address Resolution Protocol

Address Resolution Protocol ou ARP é um protocolo usado para encontrar um endereço da camada de enlace (Ethernet, por exemplo) a partir do endereço da camada de rede (como um endereço IP). O emissor difunde em broadcast um pacote ARP contendo o endereço IP de outro host e espera uma resposta com um endereço MAC respectivo. Cada máquina mantém uma tabela de resolução em cache para reduzir a latência e carga na rede. O ARP permite que o endereço IP seja independente do endereço Ethernet, mas apenas funciona se todos os hosts o suportarem.

No servidor Debian:

Execute o comando “ping” para o endereço de “broadcast”:

```
# ping -b 192.168.200.95
```

O linux por padrão não aceita pacotes icmp em broadcast: Visualize:

```
# cat /proc/sys/net/ipv4/icmp_echo_ignore_broadcast
```

Para habilitar temporariamente:

```
# echo 0 > /proc/sys/net/ipv4/icmp_echo_ignore_broadcast
```

Para habilitar permanentemente, adicione a linha a seguir no arquivo /etc/sysctl.conf:

```
# vim /etc/sysctl.conf
net.ipv4.icmp_echo_ignore_broadcast = 0
```

Salve e releia as configurações:

```
# sysctl -p
```

Execute o comando “ping” novamente para o endereço de “broadcast”:

```
# ping -b 192.168.200.95
```

Agora visualize a tabela ARP com o comando “arp” :

```
# arp  -n
```

Vamos configurar a rede IPV6 agora: Ao conectarmos o cabo de rede a placa de rede já obtém um endereço IPV6, através do endereço de sua placa de rede. Cheque o servidor Debian: Para visualizar o ipv6 configurado automaticamente:

```
# ifconfig eth0 | grep inet6
```

ou

```
# ip -6 addr show dev eth0
```

Cheque se o servidor CentOS está com o suporte a ipv6 habilitado:

```
# cat /etc/sysconfig/network:
NETWORKING_IPV6=yes
```

Verifique se a placa de rede tem suporte habilitado:

```
# cat /etc/sysconfig/network-scripts/ifcfg-eth0
IPV6INIT = yes
```

Ping as máquinas pelo IPV6:

```
# ping6 -I eth0 fe80::72f1:a1ff:fec5:fc1c
```

Onde:

-I -> especifica a interface eth0 -> é a interface para o ping6 fe80::72f1:a1ff:fec5:fc1c -> ipv6 a ser pingado

Configurando um endereço ipv6 na interface:

```
1 # ifconfig <interface> inet6 add <ipv6address>/<prefixlength>
```

No servidor Debian:

```
1 # ifconfig eth0 inet6 add ::192.168.0.1/96
```

No servidor CentOS:

```
1 # ifconfig eth0 inet6 add ::192.168.0.2/96
```

Tente pingar entre as máquinas, a partir da CentOS:

```
1 # ping -I eth0 ::192.168.0.1
```

Para remover o endereço:

```
1 # ifconfig <interface> inet6 del <ipv6address>/<prefixlength>
```

Remova de ambos: Debian:

```
1 # ifconfig eth0 inet6 del ::192.168.0.1/96
```

CentOS:

```
1 # ifconfig eth0 inet6 del ::192.168.0.2/96
```

11.7 Verificando portas abertas

11.7.1 Comando netstat

O comando netstat exibe o status das conexões, tabelas de rotas, estatísticas da interface, conexões mascaradas e participações em "multicast". Vamos ver alguns exemplos do comando.

O comando netstat sem parâmetros traduz os IP's para os nomes, por isso, desse modo, ele demora bastante para gerar um resultado. Resolve-se esse problema usando a opção "-n":

O comando exibe por padrão 4 colunas:

Proto - Protocolo que pode ser TCP, UDP, TCPv6, ou UDPv6

Local Address - Endereço local (seu PC)

Foreign Address - Endereço remoto

State - Exibe o estado da conexão de rede que podem ser CLOSE_WAIT, CLOSED, ESTABLISHED, FIN_WAIT_1, FIN_WAIT_2, LAST_ACK, LISTEN, SYN_RECEIVED, SYN_SEND, e TIME_WAIT.

Para saber mais sobre os estados das conexões consulte a RFC 793 http://tools.ietf.org/html/rfc793

-a : exibe todas as conexões e as portas TCP e UDP.

-n : exibe os números das portas ao invés do nome.

-p : exibe o PID (Process ID).

-l : exibe os sockets que estejam ouvindo.

-r : exibe a tabela de roteamento.

-t : exibe os sockets TCP.

-u: exibe os sockets UDP.

Podemos fazer uso de vários parâmetros juntos:

A opção “-n” exibe todas as portas e no formato numérico.

```
1 # netstat -n
```

As opções “-t” e “-l” exibem as conexões TCP e as portas disponíveis respectivamente para cada conexão com a sua máquina:

```
1 # netstat -ntl
```

A opção “-u” exibe as conexões UDP:

```
1 # netstat -nul
```

A opção “-p” exibe o número do processo e o nome do programa responsável.

```
1 # netstat -nltup
```

A opção “-a” exibe tanto os socket que estejam ouvindo quanto aqueles que não estejam. netstat -an

A opção “-r” exibe a rota do sistema. netstat -rn

11.7.2 Comando nmap

O Nmap é um escaneador de hosts que usa recursos avançados para verificar o estado do seu alvo. Existem diversas formas e parâmetros a serem informados durante umavarredura.

Métodos de Varredura

-sP

Ping scan: Algumas vezes é necessário saber se um determinado host ou rede está n ar. Nmap pode enviar pacotes ICMP “echo request” para verificar se determinado host ou rede está ativa. Hoje em dia, existem muitos filtros querejeitam os pacotes ICMP “echo request”, então envia um pacote TCP ACK paraa porta 80 (default) e caso receba RST o alvo está ativo. A terceira técnica envia um pacote SYN e espera um RST ou SYN-ACK.

-sR

RCP scan: Este método trabalha em conjunto com várias técnicas do Nmap. Ele considera todas as portas TCP e UDP abertas e envia comandos NULL SunRPC, para determinar se realmente são portas RPC. É como se o comando “rpcinfo -p” estivesse sendo utilizado, mesmo através de um firewall (ou protegido por TCP-wrappers).

-sS

TCP SYN scan: Técnica também conhecida como “half-open”, pois não abre uma

conexão TCP completa. É enviado um pacote SYN, como se ele fosse uma conexão real e aguarda uma resposta. Caso um pacote SYN-ACK seja recebido, aporta está aberta, enquanto um como resposta indica que a porta está fechada. Avantagem dessa abordagem é que poucos irão detectar esse scanning de portas.

- sT

TCP connect() scan: É a técnica mais básica de TCP scanning. É utilizada achamada de sistema (system call) “connect()” que envia um sinal as portas ativas.Caso a porta esteja aberta recebe como resposta “connect()”. É um dos scan mais rápidos, porém fácil de ser detectado.

-sU

UDP scan: Este método é utilizado para determinar qual porta UDP está aberta em um host. A técnica consiste em enviar um pacote UDP de 0 byte para cada porta do host. Se for recebido uma mensagem ICMP “port unreachable então a porta está fechada, senão a portapode estar aberta. Para variar um pouco, a Microsoft ignorou a sugestão da RFC e com isso a varredura de máquinas Windows é muito rápida.

-sV

Version detection: Após as portas TCP e/ou UDP serem descobertas por algum dos métodos, o nmap irá determinar qual o serviço está rodando atualmente. Oarquivo nmap-service-probes é utilizado para determinar tipos de protocolos,nome da aplicação, número da versão e outros detalhes

-O

Ativa a identificação do host remoto via TCP/IP. Irá apresentar versão do SistemaOperacional e tempo ativo

```
p<lista_de_portas>
```

Especifica quais portas devem ser verificadas na varredura. Por default, todas aspor-tas entre 1 e 1024 são varridas.

-n

Não irá resolver nome de hosts a serem varridos.

-v

Modo verbose. Mostra tudo o que está se passando.

Na máquina Debian:

Cheque o sistema operacional utilizado:

```
1 # nmap -v -n -O localhost
```

Cheque as portas abertas:

```
1 # nmap -v -sT -sU localhost
```

No servidor CentOS: Cheque em que porta o serviço ssh está escutando:

```
1 # nmap -sV localhost
```

11.8 Comando tcpdump

O comando tcpdump mostra o tráfego de uma rede. Ele exibe a descrição do con-teúdo de pacotes numa interface de rede que case com uma expressão booleana.

Sintaxe: tcpdump <opções> <dispositivo> <expressão>

Opções:

-A Imprime cada pacote em código ASCII.

-c Termina a execução após receber “n” pacotes.

-D Exibe a lista das interfaces de rede disponíveis no sistema e que o tcpdump é capaz de capturar pacotes. Esta opção associa um número a cada interface o qual pode ser usado no lugar do nome da mesma, ex: 1 - eth0, 2 – eth1, 3 – any, 4 – lo (loopback);

-i Recebe como parâmetro a interface ou o número associado a ela. Se especificado “any” captura pacotes de todas as interfaces porém, sem ser no modo promíscuo.

-n Não converte endereços em nomes. (endereços de host, número de portas, etc.)

-r Lê os pacotes a partir de um arquivo (que tenha sido criado com a opção -w).

-s Define o tamanho de cada pacote a ser capturado. É interessante utilizar o valor 1500 para que seja examinado o maior tamanho possível de pacote.

-t Não exibe o timestamp em cada linha.

-v Exibe a saída com mais detalhes.

-vv Exibe a saída com mais detalhes ainda.

-vvv Exibe a saída com informações ainda mais detalhadas.

-w Escreve os pacotes em um arquivo que pode ser lido posteriormente com a opção -r.

-x Exibe o conteúdo do pacote no formato hexadecimal.

-X Exibe o conteúdo do pacote nos formatos hexadecimal e ASCII.

expressão Seleciona quais pacotes serão exibidos. Se não for definida nenhuma expressão todos os pacotes serão exibidos. Do contrário somente os pacotes com os quais a expressão coincidir serão exibidos. A expressão consiste em uma ou mais premissas. As premissas usualmente consistem em um ID (nome ou número) precedido por um ou mais qualificadores.Existem três diferentes tipos de qualificadores:

type (tipo) Especifica host, net, port e portrange. Ex: 'host dragon', 'net 192.168', ' port 22', 'portrange 5000-5777'.

dir (direção) Indica a direção dos pacotes a serem capturados. Podem ser: src, dst, src or dst e src e dst, que significam respectivamente origem, destino, origem ou destino e origem e destino. Se nada for definido assume origem ou destino.

proto (protocolo) Define qual o tipo de protocolo será exibido. Os protocolos possíveis são: ether, fddi, tr, wlan, ip, ip6, arp, rarp, decnet, tcp e udp. Exemplos:

Acesse um site e monitore com tcpdump:

```
1 # tcpdump -i eth0 -n port 80
```

Onde:

-i -> interface, **-n ->** não resolve nomes, **port ->** identifica a porta de monitoramento.

Acesse o servidor CentOS por ssh e monitore com tcpdump:

```
1 # tcpdump -i eth0 -n port 50000
```

Serviço de Rede Telnet

Antes de partir direto para o acesso remoto via telnet, é preciso explicar uma teoria importante já que vai ser trabalhado configuração de serviços como telnet, ssh, NFS.

Os serviços de rede podem ser divididos em três tipos básicos:

- xinetd
- portmap
- stand alone

Os serviços tipo [x]inetd são aqueles que dependem do superdaemon de rede inetd – InterNET daemon. A versão mais atual do inetd é o xinetd – eXtended InterNET daemon. A função de um superdaemon é apenas controlar alguns serviços que não terão daemon próprio.

Um daemon é um processo servidor que roda em segundo plano esperando requisições. Quando se fala de inetd e xinetd, fala-se do mesmo superdaemon, mas em algumas distros (como o Debian) ainda é o inetd. Só que o xinetd possui alguns recursos a mais como controle de acesso, capacidade de fazer logs e determinar horários para que o serviço esteja disponível.

Esse superdaemon fica escutando nas portas que os serviços controlados por ele trabalham carregando o programa apropriado quando chega uma requisição na determinada porta. Exemplos de serviços tipo inetd: telnet, FTP, POP3 etc. Ao longo

do tempo, alguns dos serviços que eram controlados pelo inetd passaram a operar como stand alone a fim de contornar problemas associados ao inetd.

Os serviços stand alone são aqueles em que cada tipo de servidor possui seu daemon próprio. Esta forma de trabalho é preferida hoje em dia pois possibilita um maior controle sobre cada serviço sem separado. Exemplos de serviços que operam dessa forma: SSH, httpd (daemon do Apache), FTP, dentre outros.

Os serviços tipo portmap são aqueles que não possuem porta especifica para operar, como por exemplo, o NIS e o NFS. Estes serviços enviam uma chamada RPC – Remote Procedure Call – para a máquina servidora causando a execução de uma determina subrotina. Dessa forma quando um cliente faz a requisição de NFS a um servidor, ele está enviando um RPC tipo NFS e que quando chegar ao servidor será tratada como tal, carregando a subrotina apropriada para enviar a resposta ao cliente.

11.9 Telnet – TELetype NETwork

Telnet é um protocolo que pode ser utilizado tanto localmente quanto na internet. O telnet é considerado muito inseguro e se usar é altamente desencorajado. Ao longo dos anos vêm sendo descobertas diversas vulnerabilidades em suas implementações e, provavelmente, há muitas outras que ainda não foram determinadas.

- Por padrão, os dados enviados não são criptografados incluindo usuários e senhas

- Não possui esquema de autenticação que possibilite garantir que a comunicação está se dando entre as partes envolvidas facilitando ataques do tipo man-in-the-middle. Essas falhas têm feito cair a utilização do protocolo telnet em favor do SSH.

Atualmente, o telnet ainda é bastante utilizado para realizar configurações em equipamentos de rede especificos e também para testar o funcionamento de serviços como um servidor de POP3 eliminando a necessidade de programas clientes especializados. O telnet é controlado pelo superdaemon do Linux (inetd ou xinetd). Para que possa saber mais sobre as portas, basta dar uma olhadinha depois no seguinte arquivo:

```
# cat /etc/services
```

11.10 Instalação e configuração do Telnet

Caso queira testar o telnet, a primeira coisa é instalar o servidor e o cliente:

```
No Debian:
# aptitude install telnet telnetd openbsd-inetd
```

```
No CentOS:
# yum install telnet telnet-server xinetd
```

A configuração de um servidor telnet, na realidade, é feita configurando o [x]inetd. No Debian, o inetd é utilizado, em sistemas como o Red Hat o sistema utilizado é xinetd.

No Debian:

Verifique que a linha de configuração do telnet não está comentada no arquivo de configuração do inetd, se estiver comentada, então descomente:

```
1 # vim /etc/inetd.conf
2 telnet stream tcp nowait telnetd /usr/bin/tcpd /usr/sbin/in.telnetd
```

Descrição da opções:

- telnet: nome do servidor, como está registrado em /etc/services;
- stream: tipo de socket usado pelo protocolo, possíveis valores são: stream, dgram, raw, rdm e seqpacket; tcp tipo do protocolo usado;
- nowait/wait aguardar/não aguardar : é significativo para tipos de soquete de datagrama (dgram), outros tipos de socket usam o valor nowait.
- telnetd: usuário e grupo que irão controlar o processo de servidor;
- /usr/sbin/tcpd: é o executável para o programa TCP Wrappers
- /usr/sbin/in.telnetd: é o programa que irá lidar com as informações da conexão.

Se a linha do telnet não estiver comentada significa que o serviço já está habilitado. Para garantir que o telnet seja iniciado, reinicie o daemon do inetd. Para reiniciar o daemon do inetd:

```
1 # service openbsd-inetd stop
2 # service openbsd-inetd start
```

Faça agora uma check list para verificar se o serviço está funcionando:

```
1 # netstat -anp | grep 23
2 tcp 0 0 0.0.0.0:23 0.0.0.0:* OUÇA 2922/inetd
```

Se a porta estiver em estado de LISTEN ou OUÇA significa que ela está ouvindo, ou seja, está disponível. Você pode ver se o processo do telenet está ativo usando o fuser:

```
# fuser -v 23/tcp
23/tcp: root 2922 F.... inetd
```

Para fazer um acesso remoto em um servidor que tem o telnet habilitado é muito simples:

```
# telnet <ip_do_servidor>
```

No CentOS:

Em um sistema xinet haverá um arquivo de configuração para cada tipo de serviço e eles estarão no subdiretório xinetd.d no diretório /etc

```
# vim /etc/xinetd.d/telnet
{
  disable = no
  flags    = REUSE
  socket_type = stream
  wait     = no
  user     = root
  server     = /usr/sbin/in.telnetd
  log_on_failure  += USERID
}
```

Para reiniciar o daemon do xinetd:

```
# service xinetd restart
```

4451

Linux System Administration

www.4linux.com.br

Conteúdo

Data e Hora do Sistema e Servidor de NTP

12.1 Introdução Teórica

Manter o sistema com o horário correto é uma tarefa muito importante e que muitas vezes é negligenciada pelos administradores. Sem o horário ajustado corretamente, fica difícil agendar tarefas a serem executadas periodicamente, ou até mesmo fazer a leitura dos logs e determinar em que horário um determinado evento ocorreu. Esse detalhe torna-se ainda mais importante quando temos um servidor de e-mail rodando na máquina. Imagine um e-mail que pode ser enviado a/de qualquer parte do mundo e, somando a diferença de fuso horário, um servidor com a hora errada, fica muito difícil determinar a hora na qual o e-mail foi enviado.

Há basicamente duas formas de ajustar as configurações de horário do sistema: manualmente, utilizando os comandos “date” e “hwclock” ou usando o serviço de NTP - Network Time Protocol.

O comando **“date”** é utilizado para mostrar ou ajustar a data e hora do sistema.

Visualizar a hora:

```
1 # date
```

Acertar data e hora:

```
1 # date mmddHHMMYYYY
```

Onde:

m mês , **d** dia, **H** hora, **M** minuto, **Y** ano

EX: dia 15 de julho de 1983 às 13:15

```
1 # date 071513151983
```

Outra forma é:

```
1 # date -s "07/15/1983 13:15"
```

Para mudar somente a hora:

```
1 # date -s "12:01"
```

Para mudar somente a data:

```
1 # date -s "mm/dd/YYYY"
```

Repare que ao mudar somente a data, é alterado o horário para 00:00.

Já o comando "hwclock" é utilizado para mostrar ou ajustar a hora da BIOS da máquina sendo conhecido como RTC - Real Time Clock. Este é o relógio que fica continuamente em funcionamento mesmo que a máquina esteja desligada; de forma que

o horário esteja atualizado da próxima vez que a máquina for religada. Sua forma de utilização é bastante simples:

Visualizar hora da Bios:

```
1 # hwclock
```

Ajustar o horário da BIOS utilizando o horário do sistema:

```
1 # hwclock -w
2 Ou
3 # hwclock --systohc
```

Ajustar o horário do sistema utilizando o horário da BIOS:

```
1 # hwclock -s
2 Ou
3 # hwclock --hctosys
```

Ajustar o relógio da BIOS:

```
1 # hwclock --set --date="mm/dd/YY HH:MM:ss"
```

Outro ponto importante no que diz respeito às configurações de data e hora do sistema é a configuração da **"timezone"**, ou seja, o fuso horário em que a máquina se encontra. Essa configuração pode ser efetuada utilizando os comandos **“dpkg-reconfigure tzdata"** (específico do Debian) e **"system-config-date"** em distribuições como CentOS, Suse e Gentoo.

12.1.1 NTP -Network Time Protocol

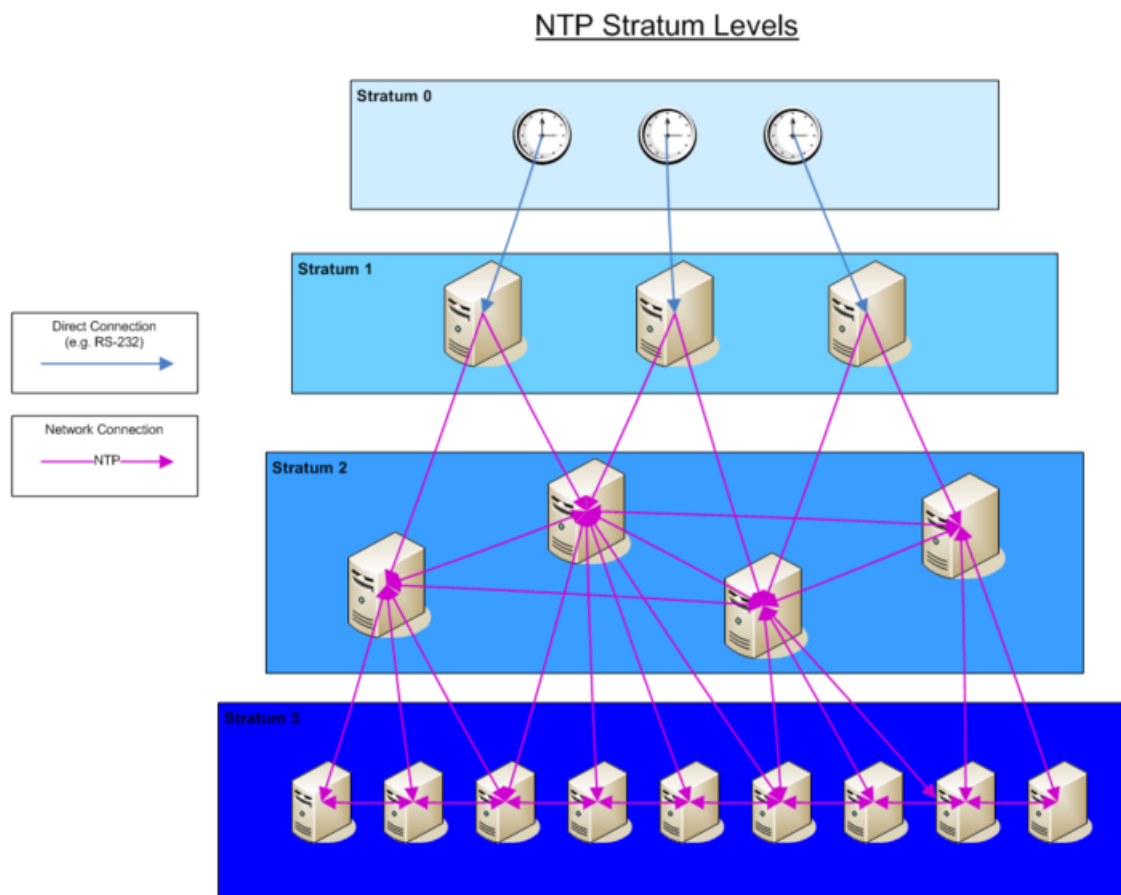
O protocolo de sincronização de horários "NTP"foi desenvolvido a fim de possibilitar que qualquer computador ligado à internet possa ajustar sua data e hora automaticamente utilizando um servidor de “hora” preciso. O NTP é um serviço na camada de aplicação que utiliza o protocolo UDP na camada de transporte fazendo uso da porta 123 para realizar a sincronização de horários.

O NTP, criado em 1985, é um dos protocolos de internet mais antigos ainda em uso e pode atingir uma precisão de aproximadamente 200µs(duzentos microssegundos).

12.1.2 Organização em Strata

A hierarquia do NTP é dividida em vários níveis, o conjunto deles é denominado "strata"e cada um deles corresponde a um "stratum". A raiz desse sistema é o denominado **“stratum 0“** e que corresponde aos relógios nucleares espalhados pelo mundo, aos quais estão conectados os servidores de **"stratum 1"**, ou seja, são eles que fazem o processamento da informação recebida do “stratum 0”.

Conectados aos "stratum 1"há o "stratum 2"que em geral estão conectados a mais de um servidor de “stratum 1” e determinam de fato qual é a hora padrão com base nos dados recebidos dos "stratum 1"utilizando o algoritmo do NTP.

Os "stratum 2"responde ao "stratum 3"que responde ao "stratum 4"e assim por diante até atingir, no máximo, 16 níveis. Uma representação esquemática dessa estrutura pode ser vista na figura.

A menos que estejamos montando um servidor para ser um "stratum 1", 2 ou 3, nunca devemos utilizar os servidores "stratum 1"ou 2 para sincronizarmos nossos servidores; mas sim acessar um "stratum 3". Dessa forma deixamos os níveis mais baixos para as máquinas que realmente precisam acessá-los.

Ainda assim, nossa política de acesso aos "stratum 3"deve ser também bastante criteriosa. Se nossa rede possuí diversas máquinas, não há sentido em fazermos todas elas se sincronizarem em um "stratum 3", mas sim escolher uma de nossas máquinas para ser um “stratum 4” e nossos clientes realizarem a sincronização a partir dela.

Sendo assim, vamos proceder com a configuração dos servidores e dos clientes.

12.1.3 Ajuste Manual de Horário

Verifique qual é a sua localização geográfica no CentOS:

```
# cat /etc/sysconfig/clock
```

Verifique qual é a sua localização geográfica no Debian:

```
# cat /etc/timezone
```

Caso esteja com um dos fusos horários incorreto, corrija-o com um dos comandos:

CentOS:

```
# rm -rf /etc/localtime
# cd /usr/share/zoneinfo/America
# ln Sao_Paulo /etc/localtime
OU
# system-config-date
```

Debian:

```
# dpkg-reconfigure tzdata
```

Verifique a data e hora do sistema e da BIOS:

```
# date
# hwclock
```

12.1.4 Trabalhando com NTP nos Clientes

Instalar o cliente NTP no Debian:

```
1 # aptitude install ntpdate
```

No CentOS o comando ntpdate já vem instalado com o pacote "ntp".

Para as máquinas da rede que forem os clientes NTP, é possível fazer a sincronização do horário com o servidor por meio do comando "ntpdate", assim:

```
1 # ntpdate [ip-do-servidor-na-rede]
```

No Debian, caso não tenha um servidor NTP na rede, podemos simplesmente reiniciar o "daemon"do "ntpdate":

```
1 # ntpdate-debian
```

O comando utilizará o(s) servidor(es) configurado(s) no arquivo /etc/default/ntpdate.

12.1.5 Configuração do Servidor de NTP

Agora que aprendemos a ajustar manualmente a hora do sistema e da BIOS, vamos utilizar o método mais preciso, ou seja, criar a estrutura de servidores e clientes de NTP.

O servidor NTP é provido pelo pacote **"ntp"**, tanto no Debian quanto no CentOS. Configure o servidor CentOS para fornecer as horas:

No CentOS verifique se o pacote ntp está instalado:

```
1 # rpm -q ntp
```

Caso não esteja instalado, instale-o:

```
1 # yum install ntp
```

Vamos configurar o nosso servidor de NTP. Abra o arquivo de configuração:

```
1 # vim /etc/ntp.conf
```

Vamos precisar obter endereços de servidores oficiais de "NTP", para isso, podemos acessar o site http://www.ntp.br, que é a página do projeto NTP. Logo na página inicial haverá uma lista de servidores públicos, mantidos pelo projeto ntp.br.

É sempre aconselhável utilizar mais de um servidor para que, caso ocorra algum erro em algum deles, o nosso sistema possa continuar com a configuração correta.

No arquivo de configuração devemos localizar a linha para configuração do servidor com o qual sincronizaremos a nossa máquina:

```
server    a.ntp.br    iburst prefer
server    b.ntp.br    iburst
server    c.ntp.br    iburst
```

Os parâmetros adicionais "iburst"fazem com que sejam enviados oito pacotes em vez de apenas um durante a sincronização inicial e o parâmetro "prefer"faz com que a resposta de um servidor preferido seja descartada se ela diferir muito das respostas dos demais servidores, caso contrário, será utilizado sem qualquer consideração para outras respostas.

Vamos adicionar a linha especificando quais "hosts"poderão realizar sincronização com a nossa máquina:

```
1 restrict 127.0.0.1
2 restrict 192.168.200.0 mask 255.255.255.0
3 disable auth
```

A primeira restrição está liberando requisições vindas do "localhost"e a segunda da nossa rede. Foi desabilitada a autenticação por chaves também. Alguns arquivos importantes no Debian:

- **statsdir /var/log/ntpstats**/ diretório onde vão ficar os logs de estatísticas do meu servidor NTP.

Os principais logs do ntp são o loopstats, que apresenta as informações do loop local, ou seja, as variáveis do sistema, e o peerstats, que apresenta as informações de cada associação.

loopstats

Seu formato é o seguinte:

day, second, offset, drift compensation, estimated error, stability, polling interval dia, segundo, deslocamento, escorregamento, erro estimado, estabilidade, e intervalo de consulta

Exemplo:

```
1 54475 73467.286 -0.000057852 31.695 0.000015298 0.006470 4
2 54475 73548.286 -0.000084064 31.688 0.000017049 0.006471 4
3 54475 73682.286 -0.000077221 31.678 0.000016130 0.006988 4
4 54475 73698.286 -0.000077448 31.677 0.000015103 0.006550 4
5 54475 73761.286 -0.000083230 31.672 0.000014275 0.006376 4
```

```
6 54475 73889.286 -0.000059100 31.665 0.000015846 0.006487 4
7 54475 74004.285 -0.000045825 31.660 0.000015548 0.006324 4
8 54475 74086.286 -0.000038670 31.657 0.000014762 0.006011 4
9 54475 74156.285 -0.000052920 31.653 0.000014699 0.005759 4
10 54475 74251.285 -0.000053223 31.649 0.000013766 0.005651 4
11 54475 74268.286 -0.000062545 31.648 0.000013292 0.005298 4
```

peerstats

Seu formato é o seguinte:

day, second, address, status, offset, delay, dispersion, skew (variance)dia, segundo, endereço, estado, deslocamento, atraso, dispersão, variância

Exemplo:

```
1 54475 34931.294 200.20.186.75 9074 0.009958844 0.008390600
    0.000390895 0.000132755
2     54475 34931.301 200.192.232.43 f0f4 0.000348814 0.015550265
        0.001120348 0.000023645
3     54475 34932.303 200.189.40.28 f0f4 0.000810708 0.017701986
        0.188995109 0.000043145
4     54475 34934.286 200.160.0.28 f0d4 0.000332344 0.000271801
        0.000620139 0.000037467
5     54475 34935.286 200.160.7.165 9614 0.000003557 0.000216088
        0.000826694 0.000022076
6     54475 34935.301 200.19.119.69 9334 0.002667663 0.015740055
        0.001858731 0.001733883
7     54475 34935.303 200.186.125.200 f034 0.004857359 0.016764821
        0.000719509 0.000303380
8     54475 34936.301 200.189.40.42 f0b4 -0.000738445 0.015510523
        0.000390094 0.000039731
9     54475 34936.301 200.19.119.120 9434 0.000009164 0.015505927
        0.000447001 0.000030319
10    54475 34938.301 200.192.232.28 f0f4 0.000363627 0.015584684
```

```
           0.063231626 0.000020460
11        54475 34939.286 200.160.0.8 f054 0.000368748 0.000334013
           0.000484437 0.000013953
12        54475 34939.286 200.160.0.43 f034 0.000325615 0.000260201
           0.000849475 0.000039634
```

A interpretação dos arquivos de log fica bastante facilitada com o uso de gráficos. E uma vez que se conhece seu formato fica muito fácil gerá-los. Várias ferramentas podem ser utilizadas para isso, mas recomenda-se o uso do **"gnuplot"**. Foge do escopo desse site dar informações detalhadas sobre a instalação ou uso dessa ferramenta, então consulte http://www.gnuplot.info/ para mais informações. O software funciona em GNU/Linux, FreeBSD, Windows e outros sistemas.

Segue um exemplo de uso, no GNU/Linux com interface gráfica, que permitirá aos interessados entender o processo:

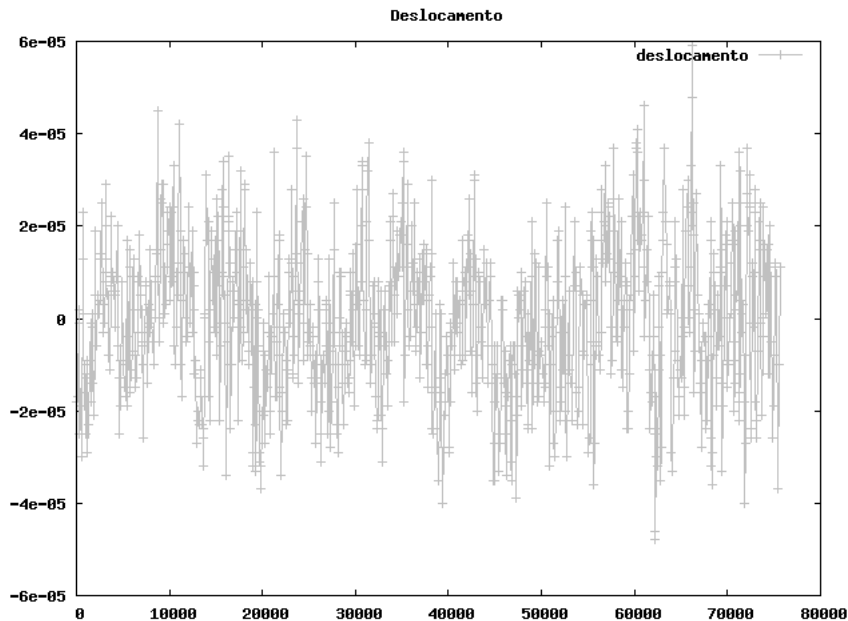
Cria-se um arquivo chamado deslocamento.txt com o seguinte conteúdo:

```
1 # set term gif
2 # set output "| display"
3 # set title "Deslocamento"
4 # plot "/var/log/ntpstats/loopstats" using 2:3 t"deslocamento" with
     linespoints lt rgb "#d0d0d0";
```

Observe-se que o comando plot faz referência ao arquivo loopstats, e usa suas colunas 2 e 3, onde: 2 representa o tempo, no dia, em segundos; e 3 representa o deslocamento, em milisegundos.

Executa-se o seguinte comando:

```
1 #   gnuplot deslocamento.txt
```

E pronto, como a saída esta redirecionada para display, vê-se o gif gerado na tela:

driftfile /var/lib/ntp/ntp.drift arquivo onde ficará configurado o valor estimado de erro de frequência entre o relógio do sistema e o servidor de sincronia de "stratum"anterior.

12.1.6 Monitorando nossa conexão NTP:

O ntp traz consigo algumas ferramentas que permitem monitorar seu funcionamento. A mais importante é o ntpq. A seguir são apresentados dois comandos do ntpq que permitem visualizar algumas variáveis importantes do ntp:

```
1 # ntpq -c pe
```

Uma reposta normal se parece com essa:

```
1 remote           refid      st t when poll reach   delay   offset
     jitter
2
```

```
3 *a.ntp.br        200.160.7.186    2 u   57   64  377    9.184
    -1.259   4.398
4 +b.ntp.br        200.20.186.76   2 u   55   64  377   18.036
    0.391   4.413
5 +c.ntp.br        200.160.7.186    2 u   51   64  377   36.675
    -0.939   3.361
```

Onde:

A primeira coluna apresenta os tally codes, que significam o seguinte: *** ->** o system peer, par do sistema, ou principal fonte de sincronização;

o -> o system peer, par do sistema, ou principal fonte de sincronização, mas apenas se a fonte for o sinal de um pulso por segundo (PPS);

+ -> candidate, ou um relógio sobrevivente, indica que é uma boa fonte de sincronização e que está sendo utilizada no momento, juntamente com o par do sistema, para ajustar o relógio local;

- -> outlyer, ou relógio afastado, indica que é uma boa fonte de sincronização, mas não sobreviveu ao algoritmo de agrupamento, ou seja, no momento há opções melhores e ela não está sendo utilizada;

x -> falseticker, ou relógio falso, indica que não é uma boa fonte de sincronização, foi descartada já no algoritmo de seleção de relógios por discordar muito das outras fontes;

. -> excess, excesso: a fonte foi rejeitada por não estar entre as 10 melhores;

(espaço em branco) -> reject, relógio rejeitado porque não responde, porque há um loop na sincronização, ou porque ele apresenta uma distância na sincronização enorme.

A seguir são apresentados os significados das demais colunas:

remote = Nome ou IP da fonte de tempo;

refid = a referência (par do sistema) à qual o servidor de tempo remoto está sincronizado;

st = o estrato da fonte de tempo;

when = quanto segundos se passaram desde a última consulta à essa fonte de tempo;

poll = de quantos em quantos segundos essa fonte é consultada;

reach = um registrador de 8 bits que vai girando para a esquerda representado na forma octal, que mostra o resultado das últimas 8 consultas à fonte de tempo: 377 = 11.111.111 significa que todas as consultas foram bem sucedidas; outros número indicam falhas, por exemplo 375 = 11.111.101, indica que a penúltima consulta falhou;

delay = atraso, ou tempo de ida e volta, em milissegundos, dos pacotes até essa fonte de tempo;

offset = deslocamento, ou quanto o relógio local tem de ser adiantado ou atrasado, em milissegundos, para ficar igual ao da fonte de tempo;

jitter = a variação, em milissegundos, entre as diferentes medidas de deslocamento para essa fonte de tempo.

Enquanto o **"ntpq -c pe"** apresenta as variáveis relacionadas a cada associação, ou seja, a cada fonte de tempo, o **"ntpq -c rl"** apresenta as variáveis (globais) do sistema.

```
1 # ntpq -c rl
```

Saída do comando **"ntpq -c rl":**

```
1 assID=0 status=0644 leap_none, sync_ntp, 4 events, event_peer/
     strat_chg,
2 version="ntpd 4.2.4p8@1.1612-o Tue Apr 19 07:08:18 UTC 2011 (1)",
3 processor="x86_64", system="Linux/2.6.32-28-generic", leap=00,
4 stratum=3, precision=-20, rootdelay=10.710, rootdispersion=18.517,
5 peer=15790, refid=200.160.0.8,
6 reftime=d25da5f5.4450ebec  Thu, Nov  3 2011 21:44:53.266, poll=6,
7 clock=d25da712.870b54e3  Thu, Nov  3 2011 21:49:38.527, state=4,
8 offset=-0.550, frequency=35.515, jitter=8.471, noise=0.604,
9 stability=0.111, tai=0
```

As mais importantes estão indicadas a seguir:

version -> versão do ntp;

stratum -> estrato do servidor local;

precision -> precisão indicada com o expoente de um número base 2;

rootdelay -> atraso ou tempo de ida e volta dos pacotes até o estrato 0, em milissegundos;

rootdispersion -> erro máximo da medida de offset em relação ao estrato 0, em milissegundos;

refid -> o par do sistema, ou principal referência;

offset -> deslocamento, quanto o relógio local tem de ser adiantado ou atrasado para chegar à hora certa (hora igual à do estrato 0);

frequency -> erro na frequência do relógio local, em relação à frequência do estrato 0, em partes por milhão (PPM).

Vamos determinar se a sincronização está funcionando corretamente. Para isso vamos alterar a hora do sistema e depois iniciar o serviço de NTP:

```
# date 010101012009

# /etc/init.d/ntpd stop

# /etc/init.d/ntpd start && tail -f /var/log/messages
```

Agora, verifique a hora do sistema novamente e veja se funcionou:

```
# date
```

Configurando o client Debian para sincronizar com o servidor:

```
# aptitude install ntp
```

Adicione o servidor a lista de servidores e comente os demais:

```
# vim /etc/ntp.conf

# adicione na linha 15 e comente os outros server:
server 192.168.200.100 iburst prefer
```

Vamos determinar se a sincronização está funcionando corretamente. Para isso vamos alterar a hora do sistema e depois iniciar o serviço de NTP:

```
# date 010101012009
# /etc/init.d/ntp stop
# /etc/init.d/ntp start && tail -f /var/log/syslog
```

Agora, verifique a hora do sistema novamente e veja se funcionou:

```
1 # date
```

12.2 Acertando horário de verão

O Decreto nº 6.558/08 (DOU de 09/09/2008), determina que o horário de verão será fixo no Brasil e começará sempre a partir da zero hora do terceiro domingo do mês de outubro de cada ano, até zero hora do terceiro domingo do mês de fevereiro do ano subsequente.

A única exceção para a regra é relacionada ao encerramento do horário de verão que, se porventura coincidir com o Carnaval, deverá ser adiado em uma semana.

12.2.1 Configurando

Verifique se existe no diretório /usr/share/zoneinfo/Brazil algum arquivo que contenha informações relativas a outros horários de verão (DICA: geralmente um arquivo com **extensão .zic**).

a) Se não existir nenhum arquivo com tais informações então crie um novo, de nome "verao.zic"por exemplo, no diretório /usr/share/zoneinfo/Brazil/. Este arquivo deverá conter as seguintes linhas:

```
1 # cd /usr/share/zoneinfo/Brazil
```

```
1 # vim verao.zic
```

```
2 Rule Brazil  2012   only  -  Oct   20    00:00 1 S
3 Rule Brazil  2013   only  -  Feb   16    00:00 0 -
4 Zone Brazil/East -3:00 Brazil BR%sT
```

Uma vez feitos os devidos ajustes no arquivo 'verao.zic' execute o comando 'zic':

```
1 # cd /usr/share/zoneinfo/Brazil/
2 # zic verao.zic
```

Neste caso em particular o comando atualizará o arquivo East.

Para verificar se as configurações corretas foram feitas, execute o comando "zdump", conforme segue abaixo:

```
1 # zdump -v /usr/share/zoneinfo/Brazil/East |grep 201[23]
```

Você deverá obter uma resposta como a que segue abaixo:

```
1 Brazil/East  Sun Oct 21 02:59:59 2012 UTC = Sat Oct 20 23:59:59 2012
      BRT isdst=0 gmtoff=-10800
2 Brazil/East  Sun Oct 21 03:00:00 2012 UTC = Sun Oct 21 01:00:00 2012
      BRST isdst=1 gmtoff=-7200
3 Brazil/East  Sun Feb 17 01:59:59 2013 UTC = Sat Feb 16 23:59:59 2013
      BRST isdst=1 gmtoff=-7200
4 Brazil/East  Sun Feb 17 02:00:00 2013 UTC = Sat Feb 16 23:00:00 2013
      BRT isdst=0 gmtoff=-10800
```

Note que em "Sat Oct 20 23:59:59 2012"o sistema ainda não está no Horário de Verão (indicação BRT). No segundo seguinte as modificações do Horário de Verão entram em vigor, adiantando o localtime em uma hora: "Sun Oct 21 01:00:00 2012

BRST"(O horário mostrado ao usuário passará para 1 da manhã, e não para meia-noite, mostrando o adiantamento do horário).

Em "Sat Feb 16 23:59:59 2012 BRST", o Horário de Verão terminará no segundo seguinte, com o localtime sendo então atrasado em 1 hora: "Sat Feb 16 23:00:00 2012 BRT"(o horário mostrado ao usuário voltará para às 23:00).

Cheque trocando a data:

```
1 # date -s '10/20/2012 23:59:30'
```

Cheque novamente:

```
1 # date -s '02/16/2013 23:59:30'
```

O horário tem de ser atualizado automaticamente quando for fazer 00:00.

Rsyslog

12.3 Introdução Teórica

A necessidade de registro das atividades dos usuários e serviços dos sistemas é notoriamente, muito importante para Administradores de Sistemas. A norma NBR ISO/IEC 27002 recomenda no item 10.10.1 as seguintes características de um sistema de logs:

1. Identificação dos usuários;

2. Datas e horários de entrada e saída de terminais;

3. Hostname ou endereço IP, para serviços acessados via rede;

4. Registro das tentativas de acessos aceitos e rejeitados.

12.3.1 Organização do Rsyslog

Cada linha do arquivo /etc/rsyslog.conf é organizada pela seguinte sintaxe:

```
# facilidade.nível  destino
```

Vamos entender o que é cada um desses itens:

facilidade - É usada para especificar que tipo de programa está enviando a mensagem.

nível - Especifica o nível de gravidade da mensagem.

destino - Especifica para onde deve ser mandada a mensagem de log.

 Vamos entender cada uma delas.

Facilidades do Rsyslog

- **auth -** Mensagens de segurança/autorização.
- **authpriv -** Mensagens de segurança/autorização (privadas).
- **cron -** Serviços de agendamento (cron e at).
- **daemon -** Outros serviços do sistema que não possuem facilidades específicas.
- **ftp -** Serviço de ftp do sistema.
- **kern -** Mensagens do kernel.
- **lpr -** Subsistema de impressão.
- **Local0-7 -** Reservados para uso local.
- **mail -** Subsistema de e-mail.
- **news -** Subsistema de notícias da USENET

.

- **security -** Sinônimo para a facilidade auth.
- **rsyslog -** Mensagens internas geradas pelo rsyslog.
- **user -** Mensagens genéricas de nível do usuário.
- **uucp-** Subsistema de UUCP.
- ***** **-** Confere com todas as facilidades.

Níveis

- **emerg -** O sistema está inutilizável.
- **alert -** Uma ação deve ser tomada imediatamente para resolver o problema.
- **crit -** Condições críticas.
- **err -** Condições de erro.
- **warning -** Condições de alerta.
- **notice -** Condição normal, mas significante.
- **info -** Mensagens informativas.
- **debug -** Mensagens de depuração.
- ***** **-** Confere com todos os níveis.

- **none -** Nenhuma prioridade.
- **error -** Sinônimo para o nível err.
- **panic -** Sinônimo para o nível emerg.
- **warn -** Sinônimo para o nível warning.

Destinos

- **arquivo -** O Rsyslog enviará os logs para um arquivo. Essa opção é a mais comum.
- **(|)-** O Rsyslog enviará os logs através de um pipe. Muito usado para redirecionar logs à um terminal.
- **(@) -** Com a arroba, o Rsyslog enviará seus logs para um computador remoto, utilizando hostname ou endereço IP.
- **user1,user2 -** Especificando o usuário, o Rsyslog enviará a mensagem para os usuários especificados. Múltiplos usuários são separados por vírgula.
- ***** - Com o asterisco, o Rsyslog enviará os logs para todos usuários logados no momento, através do comando "wall".

Arquivos importantes

Debian: Principal arquivo de log: /var/log/syslog Logs de controle do kernel: /var/log/messages Logs de depuração de “daemons”: /var/log/daemon.log

CentOS: Principal arquivo de log: /var/log/messages Logs de controle do kernel: /var/log/messages Logs de depuração de “daemons”: /var/log/messages

Debian e CentOS: Logs utilizados pelo comando “last”: /var/log/wtmp Logs utilizados pelo comando “last”: /var/log/btmp Log utilizado pelo comando “lastlog”: /var/log/lastlog Logs utilizados pelos comandos “w” e “who”: /var/run/utmp

12.4 Configurando o sistema de Logs no cliente (Debian):

1) Instale o pacote do “rsyslog” no Cliente:

```
1 Se for Debian:
2 # aptitude install rsyslog
3 OU
4 Se for CentOS:
5 # yum install rsyslog
```

2) Edite o arquivo de configuração do “rsyslog”, e ative as seguintes opções de Logs:

```
1 # vim /etc/rsyslog.conf
2 # Erros de login são enviados para o terminal 2:
3 authpriv.error      |/dev/tty2
4 # Redirecionar todos os logs para o arquivo /var/log/tudo.log:
5 *.*          /var/log/tudo.log
6 # Redirecionar a saída de logs do cron para o usuário aluno:
7 cron.*        aluno
8 # Redirecionar todos os logs para um servidor remoto Debian e Centos
    :
```

```
9  #Debian
10 *.*          @192.168.200.101
11 #CentOS
12 *.*          @192.168.200.100
```

5) Reinicie o rsyslog:

```
1 # /etc/init.d/rsyslog stop
2 # /etc/init.d/rsyslog start
```

6) Verifique o arquivo /var/log/tudo.log

```
1 # cat /var/log/tudo.log
```

7) Tente se logar como aluno no terminal 3, mas erre a senha, depois logue-se corretamente, em seguida veja o log de erro de login no terminal dois:

```
1 # ctrl + alt + 2
```

8) Reinicie o serviço do cron no terminal 1 como root e veja o log na tela do usuário aluno que está logado no terminal 3:

```
1 # /etc/init.d/cron restart
```

12.4.1 Logs Centralizados, configurando um servidor de Logs (Server Debian)

Primeiro, é necessário que o servidor seja habilitado para recebermos logs de outras máquinas, para isto, acrescente o parâmetro “-r”:

```
1 # vim /etc/default/rsyslog
```

Modifique o conteúdo do arquivo, acrescentando o parâmetro:

```
1 RYSLOGD_OPTIONS="-c4,-r"
```

 Depois, precisamos descomentar no arquivo as linhas:

vim /etc/rsyslog.conf $ModLoad imudp $UDPServerRun 514

Crie uma entrada para redirecionar os logs para um arquivo:

```
1 # vim /etc/rsyslog.conf
2 *.*    /var/log/tudo.log
```

Reinicialize o serviço Rsyslog:

```
1 # /etc/init.d/rsyslog stop
2 # /etc/init.d/rsyslog start
```

Certifique-se de que a porta está disponível para conexões remotas:

```
1 # nestat -lun | grep 514
```

Verifique o log:

```
1 # tail -f /var/log/tudo.log
```

12.4.2 Logs Centralizados, configurando um servidor de Logs (Server CentOS)

O CentOS vem com o pacote rsyslog instalado por padrão.

```
1 # vim /etc/sysconfig/rsyslog
2 #Acrescente a opção -r
3 SYSLOGD_OPTIONS="-r -m 0"
```

Crie uma entrada para redirecionar os logs para um arquivo:

```
1 # vim /etc/rsyslog.conf
2 *.*   /var/log/tudo.log
```

Reinicialize o serviço rsyslog:

```
1 # /etc/init.d/rsyslog stop
2 # /etc/init.d/rsyslog start
```

Certifique-se de que a porta está disponível para conexões remotas:

```
# nestat -lun | grep 514
```

Verifique o log:

```
# tail -f /var/log/tudo.log
```

12.4.3 Rotação de Logs

Com o tempo, os logs podem ocupar muito do espaço disponível na partição. Por isso, devemos configurar corretamente a política de rotação dos logs, ou seja, durante quanto tempo os logs serão armazenados no seu computador.

Para isso, edite o arquivo "/etc/logrotate.conf":

```
# vim /etc/logrotate.conf
# Definindo rotação de logs semanalmente
weekly
# Manter os logs de 4 semanas
rotate 4
# Criar um arquivo novo para cada rotação de logs
create
# Descomente caso queira compactar os logs em formato .gz
compress
# Todo arquivo dentro deste diretório será considerado como uma #
    configuração de log rotate.
include /etc/logrotate.d
# Configurações para wtmp e btmp
/var/log/wtmp {
missingok
monthly
```

```
16 create 0664 root utmp
17  rotate 1
18 }
19 /var/log/btmp {
20  missingok
21  monthly
22  create 0664 root utmp
23  rotate 1
24 }
25 # system-specific logs may be configured here
```

Crie uma configuração de “logrotate”:

```
1 # vim /etc/logrotate.d/errors
```

Inclua no arquivo o seguinte conteúdo:

```
1 /var/log/teste.err /var/log/teste.info {
2 daily
3 size 5M
4 sharedscripts
5      postrotate
6      /usr/bin/pkill -1 rsyslog
7    endscript
8    rotate 5
9    }
```

- **/var/log/teste.err /var/log/teste.info -** Os arquivos teste.err e teste.info serão rotacionados diariamente até 5 vezes, caso o arquivo tenha pelo menos 5M.

- **daily -** O sistema de logs será diário. size 5M - Faz a rotação somente se o arquivo alcançar 5M.

- **sharedscripts -** Marca o início do bloco de comandos.

- **postrotate -** Executa os scripts após aplicar a rotação aos arquivos.

/usr/bin/pkill -1 rsyslog - Envia sinal 1 ao processo rsyslog.

endscript - Encerra o bloco de comandos.

- **rotate 5 -** Aplica a rotação aos arquivos 5 vezes.

Adicione conteúdo aos arquivos para fazermos testes:

```
1 # cat /var/log/* >> /var/log/teste.err
2 # cat /boot/* >> /var/log/teste.info
```

Podem aparecer mensagens de erros na tela, pois o comando cat não pode visualizar o conteúdo de diretórios. Para visualizar o tamanho dos arquivos:

```
1 # du -sh /var/log/teste.*
```

Verifique que: o arquivo teste.err tem menos de 5M. o arquivo teste.info tem mais de 5M.

Agora, execute o comando “logrotate” manualmente:

```
1 # logrotate  /etc/logrotate.conf
```

Verifique que o arquivo teste.err foi rotacionado, mas o teste.info não. Isto porque o arquivo teste.info não atingiu os 5M necessários para o rotacionamento.

```
1 # ls -lh /var/log/teste*
```

Como forçar o rotacionamento dos logs:

```
1 # logrotate -f /etc/logrotate.conf
```

Repare que ao forçar o rotacionamento com a opção ”-f” todos os logs marcados para rotacionamento foram rotacionados independente do tamanho:

```
1 # ls -lh /var/log/teste*
```

No Debian o arquivo após o rotacionamento com compressão recebe a extensão: ".1.gz"No CentOS o arquivo após o rotacionamento com compressão recebe a extensão: "$date +

000

Nome do curso

www.4linux.com.br

Conteúdo

Servidor SSH

13.1 Introdução Teórica

13.1.1 SSH

Secure **Shell** ou **SSH** é o conjunto de padrões e o protocolo associado que permite estabelecer um canal seguro entre dois computadores. Ele utiliza o sistema de chave criptográfica pública para autenticar um computador remoto, podendo utilizar esse sistema de chaves, também para autenticar usuários. A idéia do SSH é prover confidencialidade e integridade dos dados trocados entre dois computadores usando criptografia e mensagens de autenticação codificadas (MACs).

Esse protocolo é tipicamente utilizado para conectar-se à máquinas remotas e executar comandos, entretanto, há inúmeras outras funcionalidades como realizar tunelamentos, redirecionamento de portas, conexões X11 (interface gráfica) além de transferência de arquivos.

Em geral, o SSH utiliza a porta 22/tcp e é a alternativa segura ao TELNET e FTP uma vez que eles não utilizam criptografia.

13.1.2 Chaves de Criptografia Assimétricas

Criar um par de chaves assimétricas tem basicamente duas funções:

- **Aumentar o nível de segurança -** “definindo uma frase senha”;
- **Facilitar a execução de scripts remotamente -** “não definir uma frase senha”.

A criação de chaves assimétricas consiste na geração de dois arquivos que contém seqüências de caracteres aleatórios (pseudo) e que só têm funcionalidade se os dois trabalharem em conjunto. Ou seja, quando criamos um par de chaves será criada uma chave pública e uma chave privada. A chave privada é sua e absolutamente ninguém deve ter acesso a ela; a sua chave pública você coloca no servidor remoto. Quando você tentar estabelecer uma conexão ela só será possível se a chave privada se encaixar na chave pública. Com esse sistema, existe apenas uma única chave privada que se encaixa em uma única chave pública.

Como só há um par que se completa, apenas quem possuir a chave privada poderá estabelecer uma conexão utilizando a respectiva chave pública. Uma ilustração do par de chaves assimétricas pode ser vista na figura: Quando criamos um par de chaves assimétricas devemos tomar o cuidado com a chave privada para que ninguém tenha acesso a ela.

13.1.3 Acesso SSH

1) Acessando uma máquina remota:

O SSH possui diversas formas de utilização; a mais básica de todas serve para estabelecer uma simples shell remota:

```
1 # ssh nome_usuario_remoto@ip_servidor
```

Ou com a opção “-l” de login:

```
1 # ssh -l nome_usuario_remoto ip_servidor
```

Outra opção é se logar no servidor remoto com o mesmo nome de usuário que você está logado, desde que este mesmo usuário exista remotamente:

```
1 # ssh ip_servidor
```

Acessar o servidor por ssh:

```
1 # ssh aluno@192.168.0.1
```

Ou:

```
1 # ssh -l aluno 192.168.0.1
```

Desconecte e conecte-se novamente sem colocar o nome do usuário:

```
1 # ssh 192.168.0.1
```

Determine qual é a porta utilizada pelo SSH:

```
1 # nmap localhost
```

A porta padrão do ssh é a porta 22.

2) Execute um comando na máquina remota:

```
1 # ssh  aluno@192.168.0.1  ls -l /etc/yum/yum.repos.d
```

13.1.4 Copiando Arquivos Remotos

13.1.5 Cópia maq_LOCAL para maq_REMOTA

Para copiar arquivo:

```
# scp arquivo usuario@ip_de_destino:/destino
```

Para copiar diretório:

```
# scp -r diretório usuario@ip_de_destino:/destino
```

Copiando arquivo:

```
# scp /home/aluno/arquivo aluno@192.168.0.1:/home/aluno
```

Copiando diretório:

```
# scp -r /home/aluno/diretorio aluno@192.168.0.1:/home/aluno
```

13.1.6 Cópia maq_REMOTA para maq_LOCAL

Para copiar arquivo:

```
# scp  usuario@ip_de_origem(remoto):/arquivo /destino
```

Para copiar diretório:

```
# scp -r usuario@ip_de_origem(remoto):/diretório /destino
```

Copiando arquivo:

```
# scp aluno@192.168.0.1:/home/aluno/arquivo /tmp
```

Visualize o arquivo copiado:

```
# ls /tmp
```

Copiando diretório:

```
# scp -r aluno@192.168.0.1:/home/aluno/diretorio /tmp
```

Visualize o diretório copiado:

```
# ls /tmp
```

13.1.7 SSH com Chaves Assimétricas

Quando criarmos o par de chaves assimétricas, será criado um diretório /.ssh na home do usuário.

Em nossa máquina local, sem ser via ssh, vamos criar o par de chaves:

Digite uma senha na passphrase, no exemplo colocamos **123456.**

```
1  # ssh-keygen -t rsa
2  Generating public/private rsa key pair.
3  Enter file in which to save the key (/root/.ssh/id_rsa):
4  Enter passphrase (empty for no passphrase): 123456
5  Enter same passphrase again: 123456
6  Your identification has been saved in /root/.ssh/id_rsa.
7  Your public key has been saved in /root/.ssh/id_rsa.pub.
8  The key fingerprint is:
9  c6:51:3e:75:0e:10:b7:98:5d:6d:81:5f:8a:8f:38:2a root@aula#
10 The key's randomart image is:
11 +--[ RSA 2048]----+
12 |        ... Eo+  |
13 |    . . o   . o . |
14 |   . . . o       |
15 |  . .   +        |
16 |. +   o S        |
17 | *   .           |
18 |. ooo            |
19 | .o+o            |
20 |   .=+           |
21 +-----------------+
```

Obs.: A passphrase pode ser desde uma senha "normal", com 6 ou 12 caracteres, até uma frase complexa, sem limite de tamanho; o importante é que não seja algo fácil de adivinhar. Caso a passphrase não seja definida o acesso remoto será sem senha.

A partir daí, ao invés de pedir sua senha, o servidor envia um "desafio"encriptado usando a chave pública. Para respondê-lo, o cliente SSH na sua máquina precisa usar a chave privada, que por sua vez precisa ser destravada usando a passphrase.

Mesmo que alguém consiga roubar sua chave privada, não conseguirá conectar sem saber a passphrase e vice-versa.

O comando gerará os arquivos ".ssh/id_rsa"e ".ssh/id_rsa.pub"dentro do seu diretório home, que são, respectivamente, sua chave privada e sua chave pública. O ".ssh/id_rsa"é um arquivo secreto, que deve usar obrigatoriamente o modo de acesso "600", para evitar que outros usuários da máquina possam lê-lo. Muitos servidores recusam a conexão caso os arquivos estejam com as permissões abertas.

1) Verifique que as chaves foram criadas:

```
1 # ls /root/.ssh
2 id_rsa  id_rsa.pub  known_hosts
```

Depois de gerar seu par de chaves, falta o comando final, que instala a chave pública no servidor, permitindo que ela seja usada para autenticação:

```
1 # ssh-copy-id -i ~/.ssh/id_rsa.pub usuario@ip_do_servidor
```

Copiando a chave:

```
1 # ssh-copy-id -i ~/.ssh/id_rsa.pub root@192.168.0.1
```

Em resumo, o que o ssh-copy-id faz nada mais é do que copiar o conteúdo do arquivo **".ssh/id_rsa.pub"**, dentro do seu diretório home, para o arquivo "**.ssh/authorized_keys**"dentro do diretório home do servidor remoto, uma operação que também pode ser realizada manualmente em caso de problemas.

Tente acessar o servidor:

```
# ssh root@192.168.0.1

Enter passphrase for key '/root/.ssh/id_rsa': 123456
Last login: Tue Jun 14 08:54:15 2011 from 192.168.0.100
```

13.1.8 Acesso por SSH sem senha com ssh-agent

O comando ssh-agent é usado para salvar as passphrases na memória, sem com isso abrir mão da segurança. Ele funciona como uma espécie de "cache", onde você digita a passphrase apenas uma vez e ela fica gravada na memória até que a sessão seja encerrada. A segurança não é prejudicada, pois a passphrase não é salva em lugar algum, fica apenas armazenada (de forma encriptada) em uma área protegida de memória, acessível apenas ao ssh-agent. Ao desligar o micro, tudo é perdido.

```
# ssh-agent
SSH_AUTH_SOCK=/tmp/ssh-dSVLR17117/agent.17117;export SSH_AUTH_SOCK;
SSH_AGENT_PID=17118; export SSH_AGENT_PID;
echo Agent pid 17118;
```

Execute os comandos que exportam as variáveis criadas pelo comando ssh-agent:

```
# SSH_AUTH_SOCK=/tmp/ssh-dSVLR17117/agent.17117; export
    SSH_AUTH_SOCK;
# SSH_AGENT_PID=17118; export SSH_AGENT_PID;
```

Adicione a chave:

```
# ssh-add
Enter passphrase for /root/.ssh/id_rsa: 123456
Identity added: /root/.ssh/id_rsa (/root/.ssh/id_rsa)
```

Tente acessar o servidor sem colocar a passphrase:

```
1 # ssh root@192.168.0.1
```

13.1.9 Configurando o servidor SSH (CentOS):

Há diversos parâmetros de configuração que podem ser alterados de forma a ajustar seus parâmetros de funcionamento.

Vamos entender alguns desses parâmetros editando o arquivo de configuração do servidor de SSH. Edite o arquivo /etc/ssh/sshd_config:

Alguns parâmetros:

Keyword Description Default AllowGroups Habilita acesso apenas para grupos especificados * AllowUsers Habilita acesso apenas para usuários especificados * DenyGroups Nega acesso apenas para grupos especificados none DenyUsers Nega acesso apenas para usuários especificados none

Port - porta de acesso ao ssh

PermitRootLogin - habilita/nega acesso do usuário root por ssh

X11Forwading - habilita/nega acesso ao X

Banner /etc/issue.net - habilita banner do issue.net

LoginGraceTime - tempo para se logar no servidor

Alterando o arquivo:

```
1  # vim /etc/ssh/sshd_config
2  # $OpenBSD: sshd_config,v 1.73 2005/12/06 22:38:28 reyk Exp $
3
4  # This is the sshd server system-wide configuration file.  See
5  # sshd_config(5) for more information.
6
7  # This sshd was compiled with PATH=/usr/local/bin:/bin:/usr/bin
8
9  # The strategy used for options in the default sshd_config shipped
      with
10 # OpenSSH is to specify options with their default value where
11 # possible, but leave them commented.  Uncommented options change a
12 # default value.
13
14 AllowUsers suporte
15 Port 2222
16 #Protocol 2,1
17 Protocol 2
18 #AddressFamily any
19 #ListenAddress 0.0.0.0
20 #ListenAddress ::
21
22 # HostKey for protocol version 1
23 #HostKey /etc/ssh/ssh_host_key
24 # HostKeys for protocol version 2
25 #HostKey /etc/ssh/ssh_host_rsa_key
26 #HostKey /etc/ssh/ssh_host_dsa_key
27 # Lifetime and size of ephemeral version 1 server key #
      KeyRegenerationInterval 1h
28 #ServerKeyBits 768
29 # Logging
30 # obsoletes QuietMode and FascistLogging
31 #SyslogFacility AUTH
32 SyslogFacility AUTHPRIV
33 #LogLevel INFO
34 # Authentication:
35 LoginGraceTime 15
```

```
36 PermitRootLogin no
37 #StrictModes yes
38 #MaxAuthTries 6
39
40 #RSAAuthentication yes
41 #PubkeyAuthentication yes
42 #AuthorizedKeysFile .ssh/authorized_keys
43
44 # For this to work you will also need host keys in /etc/ssh/
      ssh_known_hosts
45 #RhostsRSAAuthentication no
46 # similar for protocol version 2
47 #HostbasedAuthentication no
48 # Change to yes if you don't trust ~/.ssh/known_hosts for
49 # RhostsRSAAuthentication and HostbasedAuthentication
50 #IgnoreUserKnownHosts no
51 # Don't read the user's ~/.rhosts and ~/.shosts files
52 #IgnoreRhosts yes
53
54 # To disable tunneled clear text passwords, change to no here!
55 #PasswordAuthentication yes
56 #PermitEmptyPasswords no
57 PasswordAuthentication yes
58 # Change to no to disable s/key passwords #
      ChallengeResponseAuthentication yes
59 ChallengeResponseAuthentication no
60 # Kerberos options
61 #KerberosAuthentication no
62 #KerberosOrLocalPasswd yes
63 #KerberosTicketCleanup yes
64 #KerberosGetAFSToken no
65 # GSSAPI options
66 #GSSAPIAuthentication no
67 GSSAPIAuthentication yes
68 #GSSAPICleanupCredentials yes
69 GSSAPICleanupCredentials yes
70 # Set this to 'yes' to enable PAM authentication, account processing
```

```
    ,
71 # and session processing. If this is enabled, PAM authentication
      will
72 # be allowed through the ChallengeResponseAuthentication mechanism.
73 # Depending on your PAM configuration, this may bypass the setting
      of
74 # PasswordAuthentication, PermitEmptyPasswords, and
75 # "PermitRootLogin without-password". If you just want the PAM
      account and
76 # session checks to run without PAM authentication, then enable this
       but set
77 # ChallengeResponseAuthentication=no
78 #UsePAM no
79 UsePAM yes
80
81 # Accept locale-related environment variables
82 AcceptEnv LANG LC_CTYPE LC_NUMERIC LC_TIME LC_COLLATE LC_MONETARY
      LC_MESSAGES
83 AcceptEnv LC_PAPER LC_NAME LC_ADDRESS LC_TELEPHONE LC_MEASUREMENT
84 AcceptEnv LC_IDENTIFICATION LC_ALL
85 #AllowTcpForwarding yes
86 #GatewayPorts no
87 #X11Forwarding no
88 X11Forwarding yes
89 #X11DisplayOffset 10
90 #X11UseLocalhost yes
91 #PrintMotd yes
92 #PrintLastLog yes
93 #TCPKeepAlive yes
94 #UseLogin no
95 #UsePrivilegeSeparation yes
96 #PermitUserEnvironment no
97 #Compression delayed
98 #ClientAliveInterval 0
99 #ClientAliveCountMax 3
100 #ShowPatchLevel no
101 #UseDNS yes
```

```
102 #PidFile /var/run/sshd.pid
103 #MaxStartups 10
104 #PermitTunnel no
105 #ChrootDirectory none
106  # no default banner path
107 Banner /etc/issue.net
108
109 # override default of no subsystems
110 Subsystem sftp  /usr/libexec/openssh/sftp-server
```

Reinicie o serviço:

```
1 # service sshd restart
```

Agora para fazer o acesso da máquina remota ao servidor:

```
1 # ssh -p 2222 suporte@192.168.0.1
```

Onde: **-p** identifica a porta

Como usuário suporte, determine qual é a porta utilizada pelo SSH:

```
1 $ nmap -sV localhost
```

Agora ele não consegue identificar a porta do ssh, pois você alterou a porta padrão, mas ainda existem parâmetros do nmap que conseguem identificar:

```
1 $ nmap -p 0-65535 -sV localhost
```

A opção “**-p**” serve para passar um range de portas ou uma porta específica a ser escaneada.

Ele ainda não identificou qual o serviço está sendo executado na porta, então coloque a opção “**-sV**” para escanear a versão do programa:

```
1 # nmap -sV -p 0-65535 localhost
```

-s - escaneia **-V -** banner(mostra programa e versão)

13.1.10 Cópia remota com porta diferente:

```
1 # scp -P 2222 arquivo suporte@192.168.0.1:
```

-P - porta

Ao não se definir um destino após os dois pontos “:” o arquivo ou diretório será copiado para o diretório home do usuário remoto.

13.1.11 Tunelamento SSH

Uma forma simples de encriptografar protocolos que em condições normais não suportam encriptação é usar o SSH para criar túneis seguros, ligando uma das portas da sua máquina à porta do servidor onde o serviço em questão está ativo.

Por exemplo, se alguém se encontra dentro de uma instituição cuja conexão à Internet é protegida por um firewall que bloqueia determinadas portas de conexão, não será possível, por exemplo, acessar e-mails via POP3, o qual utiliza a porta 110, nem enviá-los via SMTP, pela porta 25.

As duas portas essenciais são a 80, para HTTP e a 443, para HTTPS, as quais garantem uma navegação em páginas da Web sem restrições.

Não há necessidade do administrador da rede deixar várias portas abertas, uma vez que conexões indesejadas e que comprometam a segurança da instituição possam ser estabelecidas através das mesmas.

Contudo, isso compromete a dinamicidade de aplicações na Internet. Um funcionário ou aluno que queira acessar painéis de controle de sites, arquivos via FTP ou amigos via Instant Messengers, por exemplo, não terá a capacidade de fazê-lo, uma vez que as respectivas portas para seus funcionamentos estão bloqueadas.

Para quebrar essa imposição rígida, porém necessária, o SSH oferece o recurso do **Túnel.**

Acesse em seu navegador o ip do servidor:

http://192.168.0.1

Você verá o site da dexter.com.br.

Por ssh vamos criar um túnel com esse servidor e mapear a porta 80 para a porta 12345 na nossa máquina local:

```
1 # ssh -Lporta_local:servidor_remoto:porta_remota usuá
    rio@servidor_remoto
```

Lembre-se que estamos utilizando uma porta diferente da padrão, e que o único usuário que pode se conectar é o aluno:

```
1 # ssh -p 2222 -L12345:192.168.0.1:80 aluno@IP_DO_SERVIDOR_DEXTER
```

Acesse o site no navegador localmente: http://localhost:12345

Para desconectar, deslogue do servidor.

Outras opções:

-f - O parâmetro -f"dentro do comando faz com que ele seja executado em background, liberando o terminal depois que a conexão é estabelecida.

-N - O parâmetro -N"faz com que o SSH apenas crie o redirecionamento da porta, sem abrir um terminal do servidor remoto.

13.1.12 Mensagem em broadcast: wall

O comando **wall** envia uma mensagem a todos os usuários logados no sistema. Este comando faz a leitura de um arquivo ou entrada padrão e escreve o resultado em todos os terminais onde existem usuários conectados. Somente o usuário root pode utilizar este comando.

Não confundam o comando “wall” com o antigo “netsend” da Microsoft. A mensagem “broadcast” enviada pelo “wall”, é para todos os terminais conectados naquele determinado servidor, enquanto o “netsend” faz “broadcast” para todos os endereços ativos na rede.

Enviando sua mensagem:

```
1 # wall
2 minha mensagem
3 <ctrl+d><enter>
```

1- Acesse o servidor por ssh

2- Acesse o diretório onde ficam as imagens do site da aula: /var/www/intranet/-menu

3- Renomeie o arquivo home.png para home.original.png

4- copie uma imagem do cliente para o servidor e substitua a imagem do home: arquivo a ser copiado: /usr/share/images/desktop-base/gnome-foot.png salve o arquivo com o nome de home.png

5- Acesse o site e verifique se o ícone para ir para o home mudou.

TCP Wrappers

13.2 Introdução Teórica

Os “TCP Wrappers” são usados para aplicar regras de acesso a diversos serviços em seu servidor, podendo permitir ou negar conexões a eles. Eles são controlados por dois arquivos: “/etc/hosts.allow” - configuração de acessos permitidos para determinados IPs e “/etc/hosts.deny” - configuração de acessos negados para determinados IPs. TCP - Sigla para "Transmission Control Protocol".

13.2.1 Entendendo o TCP Wrappers

Existem dezenas de possibilidades de configuração para o tcp_wrappers e você pode estudá-las em extensão através das páginas de manual “hosts_access” e “hosts_options”. Portanto, serão ilustrados apenas alguns casos interessantes do uso desta ferramenta.

As regras de controle de acesso, existentes nestes dois arquivos, têm o seguinte formato:

```
lista_de_daemons: lista_de_clientes [:comando]
```

lista_de_daemons: Lista de um ou mais nomes de daemons (como especificados no /etc/inetd.conf), ou curingas.

lista_de_clientes: Lista de um ou mais endereços ou nomes de máquinas, padrões ou curingas utilizados para especificar quais clientes podem e quais não podem acessar o serviço.

comando (opcional): É possível executar um comando sempre que uma regra casa com um padrão e é utilizada.

Veja exemplos a seguir:

Como citado anteriormente, curingas podem ser utilizados tanto na lista de daemons quanto na lista de clientes. Entre os existentes, pode-se destacar os seguintes:

ALL - Significa todos os serviços ou todos os clientes, dependendo apenas do campo em que se encontra.

LOCAL - Este curinga casa com qualquer nome de máquina que não contenha um caractere ponto ".", isto é, uma máquina local.

PARANOID - Casa com qualquer nome de máquina que não case com seu endereço. Isto geralmente ocorre quando algum servidor DNS está mal configurado ou quando alguma máquina está tentando se passar por outra.

Na lista de clientes podem ser utilizados nomes ou endereços de máquinas, ou então padrões que especificam um conjunto de máquinas. Se a cadeia de caracteres que identifica um cliente inicia com um ponto ".", um nome de máquina irá casar com este padrão sempre que o final desse nome casar com o padrão especificado. Por exemplo, se fosse utilizada a cadeia de caracteres ".minhaorganização", o nome de máquina server.minhaorganização casaria com o padrão.

Similarmente, se a cadeia de caracteres termina com um ponto ".", um endereço de máquina irá casar com o padrão quando seus campos numéricos iniciais casarem com a cadeia de caracteres especificada. Para exemplificar, se fosse utilizada a cadeia de caracteres "192.168.220.", todas as máquinas que tenham um endereço IP que inicie com estes 3 conjuntos de números irão casar com o padrão (192.168.220.0 ao 192.168.220.255).

Além destes métodos, é possível identificar um cliente através do IP/máscara de rede. Você pode especificar, por exemplo, "192.168.220.0/255.255.255.128", e qualquer máquina com endereço IP entre 192.168.220.0 e 192.168.220.127 casaria com o padrão.

13.2.2 Suporte a TCP/Wrappers

Para saber se um serviço tem suporte a "TCP/Wrappers" verifique suas bibliotecas:

```
1 # which sshd
2 # ldd /usr/sbin/sshd
```

A existência da "libwrap" confirma o suporte a "TCP/Wrappers":

```
1 libwrap.so.0 => /lib/libwrap.so.0 (0xb7ef7000)
```

Bloqueie todos os acessos ao seu servidor por "ssh":

```
1 # vim /etc/hosts.deny
2 sshd: ALL
```

Tente acessar seu servidor CentOS por ssh a partir da máquina Debian:

```
1 # ssh -p 2222 192.168.0.1
```

Não é possível devido a regra do TCP/Wrappers. Libere o acesso ssh ao seu servidor CentOS apenas para seu cliente Debian:

```
# vim /etc/hosts.allow
sshd: 192.168.0.100
```

Acesse seu servidor CentOS por ssh a partir da máquina Debian:

```
# ssh -p 2222 192.168.0.1
```

4451

Linux System Administration

www.4linux.com.br

Conteúdo

Introdução ao uso do GPG

14.1 Introdução teórica sobre criptografia

O que é criptografia?

Criptografia é o estudo dos princípios e técnicas pelas quais a informação pode ser transformada da sua forma original para outra ilegível, de forma que possa ser conhecida apenas por seu destinatário (detentor da "chave secreta"), o que a torna difícil de ser lida por alguém não autorizado.

Assim sendo, só o receptor da mensagem pode ler a informação com facilidade. A quem interessa a criptografia e quem contribuiu e contribui para ela evoluir? Militares, diplomatas e pessoas que precisam guardar segredos e até amantes.

O que é simetria?

Simetria se refere à igualdade de propriedades existente entre dois lados opostos de uma mesma situação.

Encriptar ou cifrar são termos usados para descrever a transformação de informações legíveis/úteis em dados embaralhados (sem sentido), que não se aproveita.

Decriptar ou decifrar são termos usados para descrever a transforamção de informações sem sentido (embaralhadas) em informações legíveis/úteis.

14.2 Características da criptografia simétrica

- Simples e útil;
- Cobre situações nas quais uma parte esteja envolvida;
- A pessoa que encripta é a mesma que decripta.

Exemplos de Algoritmos Simétricos:

- **(Advanced Encryption Standard)**: Está presente nos softwares BitLocker, WinZip, no padrão WPA2 etc;
- **DES**: Algoritimo criado pela IBM;
- **IDEA** :International Data Encryption Algorithm
- **Blowfish**: Elaborado por Bruce Schneier em 1993, leva o nome de um peixe que no Brasil é conhecido como baiacu

14.3 Características da criptografia assimétrica

- Proporciona privacidade e autenticidade;
- Úteis na troca de e-mail;
- Seu uso exige que cada um dos lados possua um par chaves: uma pública e outra privada

Diferença entre Chave Pública e Privada

A chave pública de um usuário de e-mail deve ser de conhecimento das pessoas que lhe desejam enviar mensagens. Você pode disponibilizar publicamente, sem problemas.

A chave privada deve sempre ser de conhecimento exclusivo do dono. Tome muito cuidado para que ninguém tome posse de sua chave privada! A par de chaves do remetente confere autenticidade, isto é, atribui legitimidade às informações trocadas entre partes diferentes. A par de chaves do destinatário confere privacidade, isto é, atribui sigilo às informações trocadas entre partes distintas.

Logo, entre os interlocutores, o uso das duas chaves leva em conta que cada um possua a chave pública do outro.

Mas afinal o que é esta tal chave?

Seria uma senha grande ou um parâmetro para se encriptar e decriptar informações.

E qual deve ser o tamanho da chave em bits?

Isto é relativo e deve ser levado em conta os seguintes fatores:

- Produtividade;
- Segurança

Chaves com 512 bits oferecem melhor desempenho nas operações criptográficas, mas são facilmente quebradas. O uso de chaves grandes causa lentidão, mas proporciona muita segurança.

Qual seria o tamanho "ideal"?

1024 bits ou 2048 bits para os mais "precavidos".

14.4 GNUPG

Vamos utililizar o software gnup. O gnupg é licenciado pela GPL e pode ser usado tanto no Linux, quanto no Windows e MAC OS X. O gnupg trabalha com critografia simétrica e assimétrica. Os algoritmos que o gnupg utiliza são:

- RSA para criptografia assimétrica;
- TripleDES, AES e Blowfish;
- Funções hash MD5 e SHA;
- ZIP, ZLIB e BZIP2 para compressão

Certifique-se que este pacote esteja instalado na sua distribuição.

Para instalar no Debian:

```
1 # aptitude install gnupg
```

Para instalar no CentOS:

```
1 # yum install gnupg
```

Para verificar a versão do GnuPG (Debian e CentOS)

```
1 # gpg --version
2 gpg (GnuPG) 1.4.10
3 Copyright (C) 2008 Free Software Foundation, Inc.
4 License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/
     gpl.html>
```

```
5  This is free software: you are free to change and redistribute it.
6  There is NO WARRANTY, to the extent permitted by law.
7
8  Home: ~/.gnupg
9  Algoritmos suportados:
10 Chave pública: RSA, RSA-E, RSA-S, ELG-E, DSA
11 Cifra: 3DES, CAST5, BLOWFISH, AES, AES192, AES256, TWOFISH,
      CAMELLIA128,
12         CAMELLIA192, CAMELLIA256
13 Dispersão: MD5, SHA1, RIPEMD160, SHA256, SHA384, SHA512, SHA224
14 Compressão: Uncompressed, ZIP, ZLIB, BZIP2
```

14.5 Gerando chaves

Acesse a maquina Debian com um usuário comum, e digite o comando abaixo para gerar a chave pública e privada.

```
1 $ gpg --gen-key
2 gpg (GnuPG) 1.4.10; Copyright (C) 2008 Free Software Foundation, Inc
     .
3 This is free software: you are free to change and redistribute it.
4 There is NO WARRANTY, to the extent permitted by law.
```

A utilizar o comando o diretório oculto .gnupg sera criado na home do usuário logado.

```
1 gpg: directory ‘/home/aluno/.gnupg’ created
```

As chaves pública e privada serão criadas com os nomes "pubring.gpg"e "secring.gpg".

```
gpg: porta-chaves ‘/home/aluno/.gnupg/secring.gpg’ criado
gpg: porta-chaves ‘/home/aluno/.gnupg/pubring.gpg’ criado
```

Varias perguntas serão feitas durante a criação do par de chaves. Responda com as seguintes opções:

Por favor selecione o tipo de chave desejado:

Tecle 2 para selecionar DSA usada para assinaturas digitais e Elgamal usada para conferir privacidade às comunicações.

DSA keys may be between 1024 and 3072 bits long.

Tecle 1024 para selecionar o tamanho da chave

Por favor especifique por quanto tempo a chave deve ser válida.

Tecle 1y para definir o tempo de renovação das chaves por 1 ano

Is this correct? (y/N)

Tecle y para confirmar a informação

You need a user ID to identify your key; the software constructs the user ID from the Real Name, Comment and Email Address in this form: "Heinrich Heine (Der Dichter) <heinrichh@duesseldorf.de>"

Nome completo: **Aluno Linux Administration**

Endereço de correio eletrônico: **aluno@dexter.com.br**

Comentário: **Maquina Debian**

Você selecionou este identificador de usuário: "Aluno Linux Administration (Maquina Debian) <aluno@dexter.com.br>"

Muda (N)ome, (C)omentário, (E)ndereço ou (O)k/(S)air? **O**

Você precisa de uma frase secreta para proteger sua chave.

Digite a frase secreta e em seguida confirme a mesma

Não há bytes aleatórios suficientes. Por favor, faça algum outro trabalho para que o sistema possa coletar mais entropia! (São necessários mais 276 bytes)

Agora o gpg fará vários cálculos para gerar uma chave pública e uma privada. Abra outro terminal com usuário root, e digite comandos de pesquisa e listagem de arquivos para gerar mais entropia

gpg: /home/aluno/.gnupg/trustdb.gpg: banco de dados de confiabilidade criado gpg: key B28CCB9D marked as ultimately trusted chaves pública e privada criadas e assinadas.

gpg: a verificar a base de dados de confiança gpg: 3 marginal(s) needed, 1 complete(s) needed, PGP trust model gpg: depth: 0 valid: 1 signed: 0 trust: 0-, 0q, 0n, 0m, 0f, 1u gpg: proxima verificação da base de dados de confiança a 2013-07-12 pub 1024D/B28CCB9D 2012-07-12 [expires: 2013-07-12] Key fingerprint = A125 D989 0EFC 74E2 6ED2 D309 3FC5 09C6 B28C CB9D uid Aluno Linux Administration (Maquina Debian) <aluno@dexter.com.br> sub 1024g/A40A66E8 2012-07-12 [expires: 2013-07-12]

14.6 Exportando uma chave pública

Para exportar a chave pública do usuário para um arquivo, utilizae o comando abaixo com o usuário comum:

```
$ gpg --armor --output "chave_publica_aluno.txt" --export "
   aluno@dexter.com.br"
```

Visualize o conteúdo do arquivo que contém a chave pública através do comando cat:

```
$ cat chave_publica_aluno.txt
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v1.4.10 (GNU/Linux)

mQGiBE//Ad0RBADlzDbqxM5Bqfqq2tq/LIVO1pYHPOvn43OoVBSvbHyvNcGb0TGX
Nw9bZ8oroYB/I5705zqLRQjuy7G090u6sMjbg6U5e9GHSTrTy6YxQ3LZ5ZftJXgO
p9awnnYv6ke2hTFhfRg53hVAL17N86QVomfNWJjEk0DKf+aeYZgIHMTvXwCgsykb
M34ZOUJbISJBW4SgcQiHTmUD/j0PeL1d0I4ekhGQcp5oENoePz8ZJNhZe/f3OTg2
qmtbeFWSqEQXNfrRXoc0eCfvIuk93TIpMYOFwUHssKIILrLwSAiat6AMcHxpFTjK
dxdbWcHNBr9OO+ddrsMVVe03jSkbX6rdd+271NcsAl+qzYN9ezUdDTv8iLacp+1o
REMtBACZ+ss+iMIU0mvnH33qoXTEz9boo+Or0LEKHqYwppZn3p5GwnBxebOAMEa/
+OqZx2oANPj0EoP0QAV1f7I2sNSf+h1v4+OP/uWCpcYpzfXmSGovcm0K0bLnycAW
IGQ90uJXQ5Ze3pW5WOmL3NFJ1olRezbcAqT5Z1qmcoAQnD6mC7RBQWx1bm8gTGlu
dXggQWRtaW5pc3RyYXRpb24gKE1hcXVpbmEgRGViaWFuKSA8YWx1bm9AZGV4dGVy
LmNvbS5icj6IaAQTEQIAKAUCT/8B3QIbAwUJAeEzgAYLCQgHAwIGFQgCCQoLBBYC
AwECHgECF4AACgkQP8UJxrKMy516YwCggwMGttnNbIqmrUYKx6NdRyMa13YAn0nM
hzMZgZmqQyvlWNn7iECMFY1ZuQENBE//Ad0QBADdQ8GM1hh9+ilWnSfoavffSMqd
1tn69uBpEmGju2f+j92qY2/wZWeiYTCYZvU6Tlfjqk6+OcozdyInpnhVXHHCodmN
HSuJKFDd2T+gG85MI4yqI95AXgHgVbOZX9U1WacMmcAIDBmVlpPunZ3E3dxijd6w
yCvWF5mOcFiFNCtoowADBgQAvTjy8D5ANTyvTb4EAz9w4J+n40zlgebUVJ45f7eU
6lwIu3m+/MCrUD8e5a3E7bqbkXPo3w48X1k4MtzgyUMhygVNaDd9b3CTcWlk6JnV
5y5yXYbU9BxABJv+TTJbwg5tHdiXcl6lFPkUeR1kdjmu1Rux6Sv7uD4siRz5m6uv
ZpOITwQYEQIADwUCT/8B3QIbDAUJAeEzgAAKCRA/xQnGsozLnQP/AKCtFY87vAfc
PeW/Oy8gsaa6igw7UgCfYMnwny2z1EeXQE/h9NssRF9dyCk=
=utVZ
-----END PGP PUBLIC KEY BLOCK-----
```

14.7 Dica de segurança

Os arquivos pubring.gpg, secring.gpg e trustdb.gpg guardam chaves e informações referentes ao par. Guarde a cópia em um local de difícil acesso.

Faça uma cópia dos arquivos do diretório /.gnupg

```
1 $ ls .gnupg/
2 gpg.conf  pubring.gpg  pubring.gpg~  random_seed  secring.gpg
     trustdb.gpg
```

Compatibilidade: O gnupg gera arquivos com extensão .gpg quando encripta. Um programa proprietário gera um arquivo com extensão .pgp.

14.8 Importado uma chave pública

Acesse a maquina CentOS com um usuário comum (diferente do aluno) e repita os procedimentos da sessão "Gerando chaves", alterando as seguintes informações:

- Nome completo
- Endereço de correio eletrônico
- Comentário

Na maquina CentOS envie para o usuário Aluno da maquina Debian, o arquivo da chave pública através do comando scp:

```
1 $ scp chave_publica_user_centos.txt aluno@192.168.200.1:
```

Na maquina Debian, importe a chave de usuário do CentOS através do comando:

```
1 $ gpg --import chave_publica_user_centos.txt
```

Para listar as chaves importadas use o comando:

```
1 $ gpg --list-keys
```

Depois de adicionar o destinatário, defina a autenticidade associada à aquela chave (o grau de confiança):

```
1 $ gpg --edit-key "Tux CentOS"
```

Use o comando "trust"para definir o grau de confiança):

```
1 Command> trust
```

Please decide how far you trust this user to correctly verify other users' keys (by looking at passports, checking fingerprints from different sources, etc.)

Tecle 5 para definir que Confia totalmente

Para sair da edição da chave use o comando quit:

```
1 Command> quit
```

14.9 Encriptar arquivos com GPG

Antes de enviar um arquivo ao usuário que importou sua chave pública, garanta a segurança na troca de informações encriptando o arquivo com o comando GPG.

Na maquina Debian, use o comando abaixo para encriptar o arquivo lista_secreta.txt usando a chave pública do usuário do CentOS:

```
1 $ gpg --recipient "Tux CentOS" --output "lista_secreta.txt.gpg" --
    encrypt "lista_secreta.txt"
```

Envie o arquivo encriptado ao usuário da maquina CentOS:

```
1 $ scp lista_secreta.txt.gpg tux@192.168.200.2:
```

14.10 Decriptar arquivos com GPG

Na maquina CentOS, use o comando abaixo para decriptar o arquivo lista_secreta.txt.gpg, informando sua frase secretao:

```
1 $ gpg --decrypt-files "lista_secreta.txt.gpg"
2 Enter passphrase:
```

Para verificar se ocorreu tudo com sucesso, liste o conteúdo do diretório

```
1 $ ls
2 lista_secreta.txt lista_secreta.txt.gpg
```

14.11 Criando assinaturas com GPG

Cenário: Vamos supor que você quer enviar um arquivo de texto e que o destinatário deseja uma prova sua de que realmente foi você quem enviou. Uma assinatura não basta, ainda mais em um texto.

Crie um arquivo chamado aviso.txt e digite uma frase:

```
1 $ echo "Sua frase aqui" > aviso.txt
```

Use o comando gpg para assinar um arquivo com a opção –clearsign:

```
1 $ gpg --clearsign aviso.txt
2 Enter passphrase:
```

Liste o conteúdo do diretório para verificar o arquivo original (.txt) e o arquivo assinado (.asc)

```
1 $ ls
2 aviso.txt aviso.txt.asc
```

Visualize o conteúdo do arquivo assinado:

```
1 $ cat aviso.txt.asc
2 -----BEGIN PGP SIGNED MESSAGE-----
3 Hash: SHA1
4 Sua frase aqui
5 -----BEGIN PGP SIGNATURE-----
6 Version: GnuPG v1.4.10 (GNU/Linux)
```

```
7 iEYEARECAAYFAkv+2
    y4ACgkQixXsqYEooo6D2QCfYSwOZE3Yy6LlXeYspLxrOdw9e1cAoKpPCAoSDYZYY/
    c8BkC9SzY+gk0P=DDzt
8 -----END PGP SIGNATURE-----
```

O arquivo "arquivo.txt.asc"está pronto para ser enviado, mas não está encriptado.

Se ele não está encriptado, falta privacidade ou autenticidade? Privacidade. Se o arquivo já está assinado, existe sim autenticidade. Agora, eu posso disponibilizar o arquivo assinado no servidor de arquivos que Maria tem acesso ou enviar por email como anexo.

Ou pode enviar o conteúdo do arquivo no corpo de uma mensagem, mas alguns clientes de e-mail tem problemas de codificação e isto pode gerar erros. O Gmail por exemplo utiliza a codificação UTF-8.

O outro usuário ao receber o arquivo testará a assinatura com o comando:

```
1 $ gpg --verify "aviso.txt.asc"
2 gpg: Signature made Thu 12 Jul 2012 21:11:54 PM BRT using DSA key ID
     8128A28E     Good     signature     from "Tux CentOS (Maquina
    CentOS)
3 gpg: <tux@dexter.com.br>"
```

4451

Linux System Administration

www.4linux.com.br

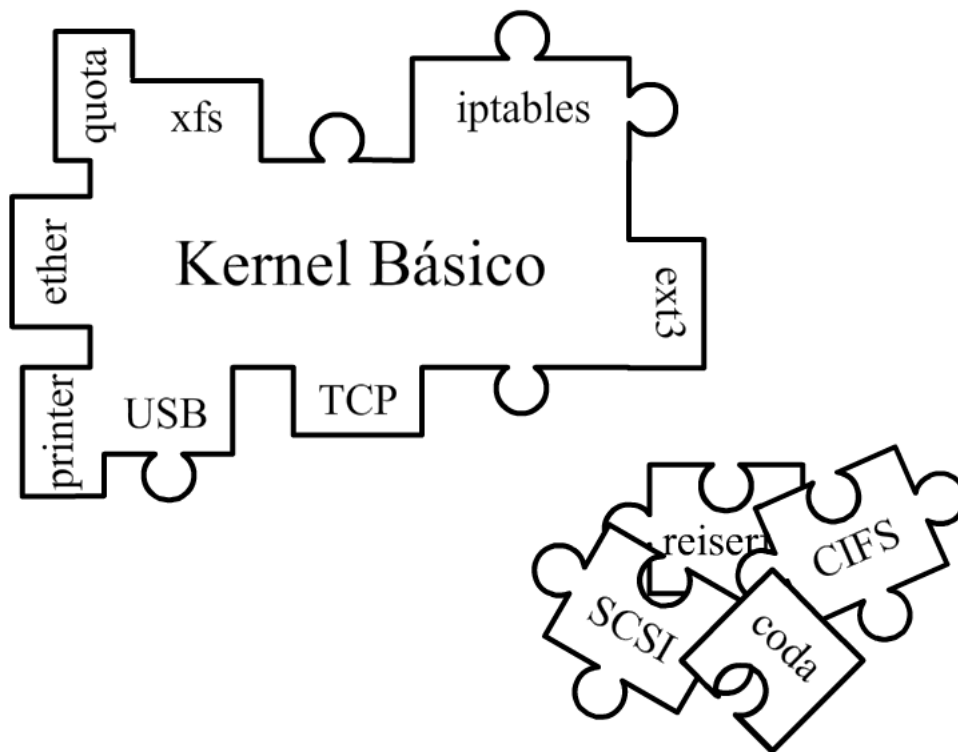
Conteúdo

Compilação do Kernel

15.1 Introdução Teórica

O centro através do qual todas distribuições são construídas é o “kernel” do sistema operacional GNU/Linux. Ele fica entre os programas de usuário e o hardware do sistema. É o “kernel” quem fornece suporte ao hardware, protocolos e alguns softwares. Vimos no capítulo de Módulos que “kerneis” genéricos baseiam-se na utilização de módulos, ou seja, o desenvolvedor compila um “kernel” básico e compila todo o resto em forma de módulos, de maneira que todos os suportes adicionais sejam adicionados de forma modular.

Quando compilamos um “kernel”, a idéia é torná-lo mais enxuto e seguro. Frequentemente seremos forçados a realizar uma compilação a fim de adicionar o suporte a alguma funcionalidade que não veio habilitada por padrão.

De certa forma, queremos passar de um esquema modular a um esquema em que a maioria, se não todas, as funcionalidades que iremos utilizar estejam “built-in” no “kernel”, ou seja, não modular, como pode ser visto na figura abaixo.

A perfeição seria ter um "kernel" bem compilado contendo apenas o conjunto de partes estritamente necessárias ao funcionamento do nosso servidor e não ter nenhum módulo externo compilado. Retiraríamos até o suporte a carregar módulos.

Configurar um "kernel" manualmente é frequentemente tido como o procedimento mais difícil que um usuário de Gnu/Linux tem que fazer. Isso não é bem verdade, depois de configurar uns dez "kerneis" você nem lembrará que foi difícil :) Como tudo na vida é mais uma questão de prática do que de inteligência.

No entanto, uma coisa é verdade: você deve conhecer muito bem seu sistema quando você começar a configurar o "kernel" manualmente. A maior parte das informações pode ser obtida utilizando o comando "lspci".

15.2 Introdução ao Kernel versão 3

A versão 3 do kernel foi lançada em 22 de Julho de 2011 em comemoração aos 20 anos do Linux, com alterações bastante técnicas incluindo suporte a mais dispositivos, redução na fragmentação no sistema de arquivos Btrfs e um backend de armazenamento para o Xen. Outras mudanças aparecem com suporte mais amplo a placas de rede wireless, webcams, e até mesmo o Kinect da Microsoft.

A versão 3.0 teve seu lançamento com 14,647,033 linhas de código, diferente de outras versões, como pode ser comparada na lista abaixo:

- **Versão 1.0.0**: 176,250 linnhas de código;
- **Versão 2.2.0**: 1,800,847 linhas de código;
- **Versão 2.4.0**: 3,377,902 linhas de código;
- **Versão 2.6.0**: 5,929,913 linhas de código;
- **Versão 3.0**: 14,647,033 linhas de código.

15.3 Funcionalidades do Kernel 3.3

Quais são as novas funcionalidades do Kernel 3.3?

A versão 3.3 do Kernel trouxe muitas novidades significativas, e é esta a versão que sera instalada em nossa pratica. Acompanhe abaixo um resumo de algumas novidades:

Suporte melhorado a sistemas de arquivos

Redimensionamento do sistema de arquivos Ext4 mais inteligente e suporte a balanceamento e re-striping ao sistema de arquivos Btrfs, permitindo que a migração de RAIDs criados com o Btrfs seja pausada, cancelada e reiniciada após uma falha.

Fusão com o projeto Android

A fusão entre o Kernel Linux e o kernel modificado do Android, se torna realidade nesta versão, trazendo melhorias para ambos os projetos onde proporciona a melhora do suporte do sistema operacional em outras plataformas ou até mesmo rodar aplicativos dele usando o Kernel Linux.

Melhor ligação em interfaces de redes

Há um novo dispositivo que combina múltiplos dispositivos Ethernet em um único dispositivo virtual. Este dispositivo de rede virtual pode usar a técnica Round-Robin para dividir o tráfego de rede entre as múltiplas portas; alternativamente, uma porta designada de "backup ativo"pode assumir a conexão caso ocorram problemas com a conexão primária de rede.

Open VSwitch

O Open vswitch é uma implementação de software de um switch de rede de múltiplas camadas, onde está sendo mesclado na árvore principal do Kernel. Esta implementação é projetada para cenários mais complexos, e especialmente para ser usado como um vswitch em ambientes de servidores virtualizados.

Suporte a drivers

Esta versão teve uma revisão melhorada no suporte do Kernel (hardware), para que houvesse progresso nos drivers de código aberto para a processadores AMD, Intel e processadores gráficos NVIDIA.

Suporte de inicialização EFI

Esta versão apresenta um esboço de inicialização EFI que permite que uma ima-

gem bzImage x86, seja carregada e executada diretamente pelo firmware do EFI. O bzImage aparece para o firmware como uma aplicação EFI. Tanto a BIOS e gerenciadores de inicialização EFI podem carregar e executar a mesma bzImage, permitindo que uma única imagem de kernel possa trabalhar em qualquer ambiente de inicialização.

15.4 Conhecendo o hardware da maquina

Vamos conhecer um pouco do hardware antes de iniciarmos a configuração do "kernel":

```
1 # cat /proc/interrupts
2 # cat /proc/ioports
3 # cat /proc/meminfo
4 # cat /proc/cpuinfo
```

Dica LPI: Lembre-se:

/proc/interrupts - contém as informações dos canais IRQ;

/proc/ioports - contém as informações "Input/Output";

/proc/meminfo - contém informações da memória;

/proc/cpuinfo - contém informações do processador;

/proc/mtab - contém informações dos diretórios que estão montados;

/proc/swaps - contém informações dos "swaps" em uso;

/proc/dma - contém informações dos canais de “DMA” em uso;

/proc/filesystems - contém informações dos sistemas de arquivos;

/proc/modules - contém informações dos módulos carregados;

Para obtermos informações a respeito dos componentes “PCI” e “USB” conectados à máquina devemos instalar dois programas o “lscpi “e o “lsusb”:

```
1 # aptitude install pciutils usbutils
```

```
1 # yum install pciutils usbutils
```

Veja os dispositivos PCI e USB conectados à máquina:

```
1 # lspci
2 # lsusb
```

15.5 Configurar, compilar e instalar o Kernel

Agora que você sabe as novidades do Kernel 3.3 e tem conhecimento do hardware da maquina, vamos colocar em pratica a implementação da versão 3 do Kernel. O download sera feito no endereço do kernel.org, e a distribuição usada sera o Debian 6.

Antes de compilar qualquer versão do Kernel, comece instalando os pacotes abaixo para resolver as dependências durante a compilação.

```
1 # aptitude install make gcc g++ autoconf libncurses5 libncurses5-dev
     ncurses-base ncurses-bin ncurses-term
```

```
1 # yum install ncurses-devel
2 # yum groupinstall "Development Tools"
```

O proximo passo para compilarmos um “kernel” é fazer o download de seu código fonte a partir do site:

Seguindo a FHS acesse o diretório /usr/src para baixar a fonte do Kernel.

```
1 # cd /usr/src/
```

Use o comando wget e baixe a versão 3.3.3 usando o endereço abaixo:

```
1 # wget -c http://www.kernel.org/pub/linux/kernel/v3.0/linux-3.3.3.
     tar.bz2
```

Para desempacotar e descompactar use o comando tar no arquivo baixado

```
1 # tar jxvf linux-3.3.3.tar.bz2
```

Crie um link simbólico padronizando o nome linux como fonte do Kernel

```
1 # ln -s /usr/src/linux-3.3.3 /usr/src/linux
2 # cd linux
```

Um passo extremamente importante antes de configurar o nosso “kernel” é sempre adicionar uma EXTRAVERSION afim de organizar uma eventual estrutura de módulos no “/lib”.

Utilize o comando date para gerar uma “string de extraversion”:

```
1 # date +"-%Y%m%dc1"
2 -20120725c1
```

Adicione essa “string” à variável “EXTRAVERSION” na “Makefile” do “kernel”:

```
1 # cd /usr/src/linux-3.3.3
2 # vim Makefile
3 EXTRAVERSION = -20120725c1
```

Veja as opções da “Makefile” do “kernel”:

```
1 # make help
```

Maneiras de configurar o kernel:

- **make menuconfig**: Escolha de opções usando interface ncurses;
- **make config**: Configuração a base de perguntas e respostas no terminal;
- **make xconfig**: Configuração modo gráfico feito em QT ;
- **make gconfig**: Configuração modo gráfico feito em GTK;
- **make localmodconfig**: Configuração a base de perguntas e respostas no terminal, onde é possível retirar módulos não utilizados do seu kernel!

Se essa não for a primeira compilação desse “kernel”, é sempre recomendado realizar uma limpeza no diretório do fonte do “kernel”:

```
1 # make mrproper
```

mrproper -> remove todos arquivos gerados pelo comando make + arquivo config + vários arquivos de backup

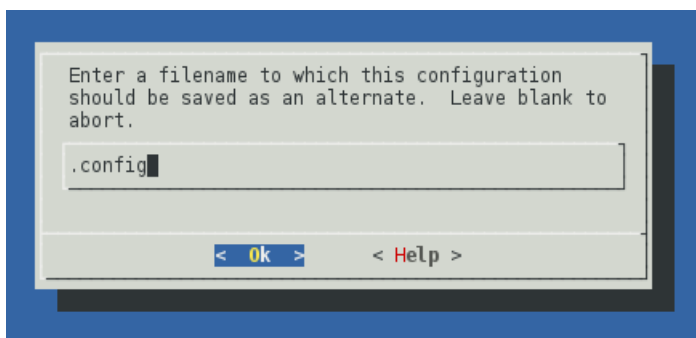
Inicie o processo de configuração do “kernel”:

```
1 # make menuconfig
```

A interface de configuração fornece uma estrutura de menus e sub-menus nos quais devemos navegar afim de selecionar as opções que desejamos adicionar, remover ou compilar como módulos.

```
.config - Linux/i386 3.3.3 Kernel Configuration
                     Linux/i386 3.3.3 Kernel Configuration
  Arrow keys navigate the menu.  <Enter> selects submenus --->.  Highlighted letters
  are hotkeys.  Pressing <Y> includes, <N> excludes, <M> modularizes features.  Press
  <Esc><Esc> to exit, <?> for Help, </> for Search.  Legend: [*] built-in  [ ] excluded
  <M> module  < > module capable
            ^(-)
                Bus options (PCI etc.)  --->
                Executable file formats / Emulations  --->
            -*- Networking support  --->
                Device Drivers  --->
                Firmware Drivers  --->
                File systems  --->
                Kernel hacking  --->
                Security options  --->
            -*- Cryptographic API  --->
            -*- Virtualization  --->
                Library routines  --->
            ---
                Load an Alternate Configuration File
                Save an Alternate Configuration File

                  <Select>    < Exit >    < Help >
```

Após salvar as configurações, podemos visualizar o arquivo gerado e copiá-lo para um lugar seguro:

```
1 # less .config
```

```
2 # cp .config /boot/config-3.3.3-20120725c1
```

Agora que está tudo pronto, vamos compilar o kernel:

```
1 # make CONFIG_DEBUG_SECTION_MISMATCH=y
```

Dica: Caso seu processador possua varios núcleos otimize a compilação através da flag “-j”. Exemplo:

```
1 # make CONFIG_DEBUG_SECTION_MISMATCH=y -j4
```

-j4: Executa a compilação do kernel em quatro processos simultâneos!

Após compilar o “kernel” e seus módulos, vamos copiar os módulos para o diretório apropriado em “/lib/modules”:

```
1 # make modules
2 # make modules_install
```

Dica LPI: O Processo de compilação do “kernel”: make ; make modules_install

Depois de compilado o “kernel”, será gerado um arquivo da imagem (o bzImage) no diretório “/usr/src/linux-3.3.3/arch/**XXX**/boot”.

Onde **XXX** é a arquitetura da máquina.

Copie-o para o diretório “/boot”, com o nome de “vmlinuz”, este é o nome dado ao kernel:

```
1 # cd arch/i386/boot/
2 # file bzImage
3 # cp bzImage /boot/vmlinuz-3.3.3-20120725c1
```

Se o nosso kernel tiver sido compilado com módulos que sejam extremamente necessários durante o boot, será necessário criar uma "imagem de boot".

Para isso, precisamos instalar o seguinte pacote no Debian:

```
1 # aptitude install initramfs-tools
```

Agora devemos construir nosso arquivo "initrd" no "/boot":

```
1 # mkinitramfs -o /boot/initrd.img-3.3.3-20120725c1 /lib/modules
    /3.3.3-20120725c1
```

```
1 # mkinitrd /boot/initrd-3.3.3-20120725c1.img /lib/modules
    /3.3.3-20120725c1
```

Feitos esses procedimentos devemos configurar o nosso “Boot Loader”.

Esta é parte mais simples! Se o Kernel tem o nome vmlinuz-versão no diretório /boot. Aos invés de criar uma configuração manual com a entrada do Kernel e initrd-img, apenas digite o comando abaixo e a entrada no Grub 1 ou 2 sera feita de forma automática:

```
1 # update-grub
```

4451

Linux System Administration

www.4linux.com.br

Conteúdo

Gerenciadores de Boot

16.1 Introdução Teórica

Um “bootloader” é o software responsável por carregar o sistema operacional durante a inicialização do sistema. Há vários “bootloaders” diferentes disponíveis no GNU/Linux. O papel do “bootloader” é fornecer uma lista de opções de sistemas operacionais disponíveis na máquina e que podem ser carregados. Uma vez que o usuário. Escolheu qual sistema deseja “subir” o GRUB inicia o carregamento do “kernel” na memória “RAM” o qual passa a ter o controle sobre a máquina.

Ao contrário da maioria dos programas que colocam seus arquivos de configuração no diretório “/etc” o “grub” coloca-os no diretório “/boot/grub”.

16.1.1 GRUB (Padrão CentOS)

Este é o BootLoader padrão do CentOS. Para que nosso novo “kernel”, possa ser inicializado, devemos configurar nosso “bootloader”. Veja agora como fazer isso no “Grub”.

Embora seja "enjoado", o grub não é tão complicado como pode parecer à primeira vista. Vamos aproveitar a deixa para aprender um pouco mais sobre ele.

O grub usa o arquivo de configuração **"/boot/grub/menu.lst"** no Debian, no CentOS ele é um link para **“/boot/grub/grub.conf”**. Este arquivo é lido a cada boot, por isso

não é necessário reinstalar o grub ao fazer alterações, como no caso do lilo.

Este é um exemplo de arquivo de configuração do grub:

```
1  default 0
2  timeout 5
3  color cyan/blue white/blue
4  splashimage=(hd0,0)/grub/splash.xpm.gz
5
6  title CentOS
7  root (hd0,0)
8  kernel /vmlinuz-2.6.32-71.el6lx86_64 ro root=/dev/VolGroup/lv_root
9  initrd /initramfs-2.6.32-71.el6lx86_64
10
11 title Microsoft Windows
12 root (hd0,2)
13 makeactive  # Se usar Windows7 comente esta linha.
14 chainloader +1
15
16 title Debian Squeeze
17 root (hd0,3)
18 kernel /vmlinuz-2.6.31 ro root=/dev/sda3
19 initrd /initrd.img-2.6.31
```

O CentOS é o default, por causa da opção "default 0"no início do arquivo. Do ponto de vista do grub, o CentOS é o sistema "0", o Windows é o sistema “1”, enquanto que o Debian é o sistema “2”. Note que ele conta os sistemas incluídos na lista a partir do zero.

Se você quisesse que o Debian passasse a ser o sistema default, bastaria trocar "default 0"por "default 2". O mesmo vale para os outros sistemas operacionais instalados.

A linha "timeout 5"é um pouco mais cosmética. Ela diz que se você não pressionar nenhuma tecla na tela de boot, o sistema default será iniciado depois de 5 segundos.

Você pode aumentar ou diminuir o tempo a seu gosto.

A linha "color cyan/blue white/blue"também é cosmética. Ela apenas indica as cores do texto e do fundo na tela de boot. Veja que as cores são definidas duas vezes. Da primeira você diz as cores que são usadas quando é exibida a mensagem de boot e na segunda as cores que serão usadas em micros onde não seja possível exibir a imagem de fundo.

Completando, temos a linha "splashimage=(hd0,0)/grub/splash.xpm.gz", que indica a imagem de fundo que será exibida. No caso do CentOS é usado um arquivo de tema, que é composto por diversos arquivos dentro da imagem de fundo.

O "(hd0,0)"dentro da opção diz a partição onde o CentOS está instalado, onde ele vai procurar o arquivo. Como pode ver, o grub usa uma nomenclatura própria para designar as partições do HD, o que acaba sendo o aspecto da configuração mais difícil de entender.

No Linux os HDs e partições são acessados através de dispositivos especiais, localizados dentro do diretório "/dev". Um HD IDE instalado como master na primeira porta IDE, é visto pelo sistema como "/dev/hda"e a primeira partição dentro dele é vista como "/dev/hda1". Se você usasse um HD serial ATA, então ele seria visto como "/dev/sda"e a primeira partição como "/dev/sda1".

Se você está me acompanhando até aqui, sente e respire fundo, pois nada disso vale para o grub. Para "simplificar", os desenvolvedores decidiram adotar uma nomenclatura própria, onde os HDs e partições são nomeados a partir do zero.

Ou seja, o "/dev/hda1"ou “/dev/sda1” é referenciado na configuração do grub como "(hd0,0)"(primeiro HD, primeira partição). O "(hd0,2)"do exemplo seria referente à terceira partição do primeiro HD, ou seja, faria referência ao "/dev/hda3"ou “/dev/sda3”.

Em resumo, na nomenclatura adotada pelo grub temos:

```
1 /dev/hda = 0
```

```
2 /dev/hdb = 1
3 /dev/hdc = 2
4 /dev/hdd = 3
```

As partições dentro de cada HD são também nomeadas a partir do zero:

```
1 /dev/hda1 ou /dev/sda1 = 0,0
2 /dev/hda2 ou /dev/sda2 = 0,1
3 /dev/hda3 ou /dev/sda3 = 0,2
4 /dev/hda4 ou /dev/sda4 = 0,3
5 /dev/hda5 ou /dev/sda5 = 0,4
6 /dev/hda6 ou /dev/sda6 = 0,5
```

Para o grub esta distinção entre hds não existe. O "/dev/sda1"continua sendo "(hd0,0)"dentro do grub.

O que acontece então se você tiver um HD IDE e outro SATA na mesma máquina? Bem, aí depende de como eles estiverem configurados dentro do setup. O HD "primário", ou seja, o que o BIOS acessa primeiro na hora de carregar o sistema, será visto como "(hd0)", independentemente de ser SATA ou IDE, enquanto o outro será visto como "(hd1)".

Uma forma de confirmar isso é checar o conteúdo do arquivo **"/boot/grub/device.map"** (com o sistema já instalado). Ele contém uma lista dos HDs detectados pelo grub, e o endereço atribuído a cada um.

Agora que entendemos como o grub nomeia os HDs e partições, podemos ir ao que interessa, ou seja, entender como funcionam as múltiplas seções do grub, que permitem carregar cada sistema operacional.

No exemplo, o HD está configurado da seguinte forma:

```
1 /dev/sda1: CentOS
```

```
2 /dev/sda2: Windows
3 /dev/sda3: Debian
```

Esta configuração vem bem a calhar, pois permite explicar os casos mais comuns, ou seja, a seção referente ao CentOS, ao Windows e referente a outras distribuições Linux, no caso o Debian.

Vamos começar com a seção do CentOS:

```
1 title CentOS
2 root (hd0,0)
3 kernel /vmlinuz-2.6.32-71.el6lx86_64 ro root=/dev/VolGroup/lv_root
4 initrd /initramfs-2.6.32-71.el6lx86_64
```

A linha **"title"** contém apenas o nome do sistema, da forma como ele irá aparecer na tela de boot. Não é preciso que o nome indique corretamente o sistema, você pode usar apelidos, o importante é apenas que um sistema receba um apelido diferente do outro.

A linha **"root"** logo a seguir, indica a localização do /boot (no formato do grub), ou seja, onde o sistema está instalado. Como o CentOS neste caso está instalado na primeira partição do HD, usamos "(hd0,0)".

A terceira linha, **"kernel"**, indica o arquivo com o kernel, que será carregado no início do boot. O Kernel vai sempre dentro da pasta "/boot"e o arquivo tem o nome padrão de "vmlinuz", seguido da versão, como "vmlinuz-2.6.32-71.el6lx86_64". Além de indicar a localização do arquivo, você pode incluir opções que serão passadas para ele no início do boot, por exemplo: "acpi=off", "vga=791"e assim por diante.

A opção de acpi=off desabilita o gerenciador de energia acpi, já a opção vga=791 indica a resolução de vídeo que será usada no terminal. O número "791"indica 1024x768, "788"indica 800x600 e se você substituir o número pela palavra "normal", o terminal passa a usar a resolução de texto padrão, como nos monitores CGA ;).

Usando "vga=normal"o bootsplash exibido durante o carregamento do sistema também deixa de funcionar.

Finalmente, temos a linha **"initrd"**, que é opcional, permitindo indicar a localização de um arquivo initrd, que será carregado junto com o Kernel. O initrd nem sempre é usado. Quando necessário, ele é gerado durante a instalação, incluindo módulos de que o sistema precisará no início do boot. Se ele não estiver dentro da pasta "/boot"junto com o Kernel, não precisa se preocupar, pois ele não está sendo usado.

Em seguida temos a seção referente ao Windows:

```
1 title Microsoft Windows
2 root (hd0,2)
3 makeactive
4 chainloader +1
```

O Windows é um caso especial, pois ele não é carregado diretamente pelo grub. Em vez disso ele é inicializado em um modo chamado de "chainload", onde o grub simplesmente carrega o gerenciador de boot do Windows (que é instalado dentro da partição) e deixa que ele se encarregue de inicializar o sistema. Isso é indicado pela linha **"chainloader +1"**.

Isto simplifica as coisas, pois você precisa apenas indicar um nome ou apelido na linha "title"e indicar a partição onde ele está instalado na linha "root". No nosso exemplo, o Windows está instalado na terceira partição do HD, por isso o "(hd0,2)".

A opção **"makeactive"** marca a partição do Windows como ativa, uma configuração que é necessária ao inicializar o Windows 95/98/ME, onde ainda é utilizado o MS-DOS na fase inicial do boot. Ela não é mais necessária no XP ou no Vista, mas, como também não atrapalha, é comum que ela continue sendo usada.

Concluindo, temos a seção referente ao Debian, que pode ser usada (com as devidas modificações) também para outras distribuições Linux instaladas no HD:

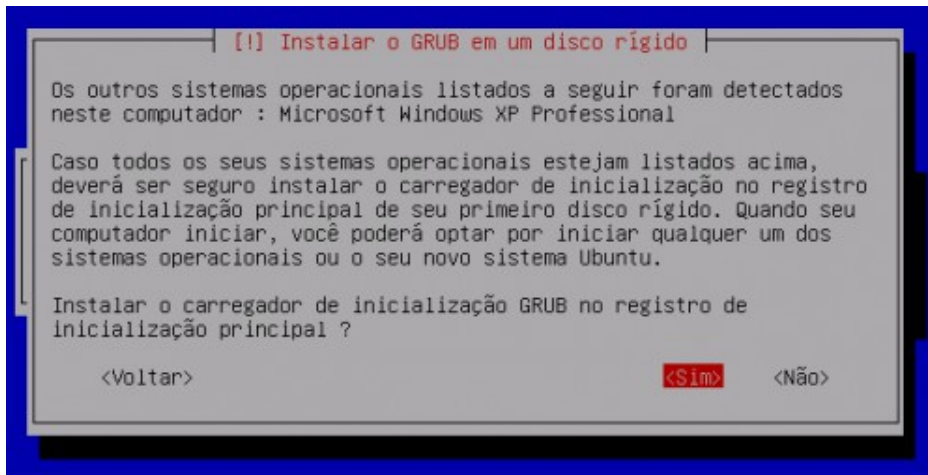
```
title Debian GNU/Linux (testing/unstable)
root (hd0,3)
kernel /boot/vmlinuz-2.6.18 ro quiet vga=791
initrd /boot/initrd.img-2.6.18
```

Esta seção é muito similar à seção do CentOS, que vimos a pouco. O Debian neste caso está instalado na quarta partição, que no grub é vista como "(hd0,3)". O importante é indicar corretamente o arquivo com o Kernel, dentro da partição e o initrd (caso exista).

Você poderia duplicar estas quatro linhas para incluir outras distribuições. Bastaria alterar a partição dentro da linha "root"e indicar corretamente o arquivo do Kernel e o initrd que seriam usados.

Embora seja um tema um pouco chato de estudar, é importante conhecer pelo menos o básico sobre a configuração do grub, pois ele é usado atualmente em praticamente todas as distribuições, de forma que é um conhecimento que você vai acabar usando bastante.

Continuando, a configuração feita no arquivo "/boot/grub/menu.lst"é lido pelo grub a cada boot, por isso você não precisa se preocupar em regravar o grub a cada alteração no arquivo, como no caso do lilo. Em geral, você só precisa regravar o grub em caso de acidentes, como quando ele é subscrito acidentalmente, ao reinstalar o Windows ou instalar outra distribuição no mesmo HD.

No caso das distribuições, Linux, quase sempre você tem a opção de instalar gravando o gerenciador de boot na partição, o que permite reinstalar sem subscrever o grub "titular". Aqui, por exemplo, temos um screenshot do instalador do Ubuntu:

O problema é o Windows, já que ele chega "chutando o balde", gravando seu gerenciador de boot na MBR sem nem te consultar.

Isto acaba se revelando um grande problema, já que você perde o acesso ao Linux instalado no HD sempre que precisar reinstalar o Windows.

Nestes casos, você pode regravar o grub dando boot com um live CD do linux.

Dê boot pelo CD e abra um terminal como root. A partir daí, use o comando "grub"para entrar no prompt do grub, onde usaremos os comandos para regravar o gerenciador de boot:

```
1 # grub
```

Dentro do prompt, precisamos rodar dois comandos, especificando a partição onde o CentOS (ou a distribuição "dona"do grub) está instalada e o dispositivo onde o grub será instalado.

Comece rodando o comando "root", que especifica a partição de instalação do sistema. No exemplo, o CentOS está instalado no "(hd0,0)", de forma que o comando fica:

```
1 grub> root (hd0,0)
```

Falta agora o comando "setup", que especifica aonde o grub será gravado. Neste caso, estou gravando o grub na MBR do primeiro HD:

```
1 grub> setup (hd0)
```

Terminando, você pode sair do prompt do grub usando o "quit"e reiniciar o micro. Este é um exemplo de operação que é mais simples no grub. No lilo, era necessário montar a partição e abrir um chroot para conseguir regravar o gerenciador :).

Mais um problema comum acontece quando você precisa configurar o grub numa máquina com vários HDs. Nestes casos, além de verificar como o grub detectou cada um, você precisa se preocupar em gravar o grub no MBR do HD correto.

O problema é muito simples. Quando você possui mais de um HD na máquina, você configura uma ordem de boot no Setup do micro. O HD que estiver em primeiro na ordem de boot do setup, será usado para inicializar a máquina e, consequentemente será reconhecido pelo grub como "(hd0)".

Se você instalar o CentOS no segundo HD, e o grub for instalado na MBR do segundo HD, o CentOS não vai inicializar depois de instalado, pois o BIOS do micro continuará lendo o MBR do primeiro HD.

A solução no caso é bem simples. Mesmo que você instale o CentOS, ou qualquer outra distribuição no segundo HD, tome sempre o cuidado de gravar o grub no MBR do primeiro HD. Se você está instalando o Debian (por exemplo), na partição /dev/sdb1 (a primeira partição do segundo HD) o "root", ou seja, o dispositivo aonde o sistema está sendo instalado será "(hd1,0)", mas na hora de gravar o grub, você indicaria o "(hd0)", que é o primeiro HD.

Ao fazer isso manualmente pelo prompt do grub, você usaria os comandos:

```
# grub
grub> root (hd1,0)
grub> setup (hd0)
grub> quit
```

Note que isto é necessário apenas ao regravar o grub manualmente. Outra pegadinha é que quando você tem uma instalação do Windows no segundo HD (hd1,0 no grub), como em situações onde você compra outro HD para instalar Linux e instala o HD com o Windows como secundário, é necessário adicionar duas linhas na seção do grub referente ao Windows. Elas fazem com que a posição lógica dos dois HD seja trocada, fazendo com que o Windows pense que está inicializando a partir do primeiro. Sem isso, você tem um erro de "partição inválida"durante o boot e o Windows não é carregado.

Ao adicionar as duas linhas, a seção referente ao Windows ficaria:

```
title Windows
root (hd1,0)
makeactive
chainloader +1
map (hd1) (hd0)
map (hd0) (hd1)
```

16.1.2 Segurança no grub

Para melhorarmos nossa segurança local, uma boa seria colocar senha no grub, então vamos fazer melhor, iremos colocar uma senha criptografada nele:

```
# /sbin/grub-md5-crypt
$1$q0ZwgzZO$BMy4amrK53Q01oRLg.W166ivy
```

Após digitarmos nossa senha ela é criptografada em md5, aí basta copiá-la e editar o arquivo /boot/grub/menu.lst.

Adicione no começo do arquivo a seguinte linha:

```
1 password --md5 \$1\$q0ZwgzZO\$BMy4amrK53Q01oRLg.W166ivy
```

A importância de termos senha no Grub é que se o mesmo estiver livre de senha, qualquer pessoa na hora da inicialização pode editá-lo e inicializá-lo para ganhar poderes de root sem saber a senha.

16.2 GRUB2 (Padrão Debian)

Este é o bootloader padrão no Debian Squeeze. Na inicialização do computador o GRUB2 apresenta o menu e espera a atuação do usuário dentro do tempo fixado ou transfere automaticamente o controle para o sistema operacional.

GRUB2 é um software Open Source. Ele é descendente do GRUB (GRand Unified Bootloader).

Foi completamente reescrito para dar ao usuário flexibilidade e performance significativamente aumentadas.

16.2.1 Melhorias

As melhorias em relação ao GRUB incluem :

- apoio de scripts - módulo de carregamento dinâmico - modo de recuperação - menus personalizados - temas - suporte ao menu gráfico - imagens iso de Live CD de

boot diretamente do disco rígido - nova estrutura de arquivos de configuração - sem suporte a plataforma x-86 (tal como PowerPc) - suporte universal a UUIDs

16.2.2 GRUB vs GRUB2

O menu do GRUB2 parecerá familiar aos usuários do Grub mas há muitas diferenças internamente.

- pressione a tecla SHIFT para mostrar o menu durante o boot (no GRUB legacy formalmente ESC). - /boot/grub/menu.lst do GRUB legacy foi substituido por /boot/-grub/grub.cfg no GRUB2.

- não há "/find boot/grub/stage1"no prompt do grub. Stage 1.5 foi eliminado

- o principal arquivo de menu /boot/grub/grub.cfg não é para ser editado mesmo pelo “root”.

- grub.cfg é sobrescrito sempre que houver uma atualização, um kernel for adicionado/removido ou o usuário executar update-grub.

- o usuário pode criar um arquivo personalizado em /etc/grub.d/40_custom com suas próprias entradas. Esse arquivo não será sobrescrito.

- o arquivo de configuração principal para alterar as configurações do menu é o /etc/default/grub.

- existem vários arquivos para configurar o menu - /etc/default/grub mencionado acima e todos os arquivos da pasta /etc/grub.d/.

- mudou a numeração das partições. A primeira partição agora é 1 em vez de 0. O primeiro dispositivo no entanto continua 0 (não mudou).

- buscas automáticas para outros sistema operacionais como Windows sempre que update-grub é executado.

- nenhuma mudança na configuração dos arquivos acontecerá até que o comando update-grub seja executado.

16.2.3 Hierarquia de arquivos e diretórios

/boot/grub/grub.cfg

É o que mais se assemelha ao **/boot/grub/menu.lst** do GRUB mas diferentemente desse o grub.cfg não se destina a ser editado. Cada seção é claramente delimitada com "(### BEGIN)" e a referência do arquivo na pasta **/etc/grub.d** a partir da qual a informação foi gerada. **grub.cfg** é atualizado executando o comando update-grub e é automaticamente modificado quando há uma atualização ou instalação/remoção de kernel.

Por padrão, e sempre que o comando update-grub é executado, este arquivo é feito "somente leitura". Isto porque a intenção é que o arquivo não seja editado manualmente.

O usuário também verá uma infinidade de arquivos *. mod na pasta /boot/grub. Esses arquivos são da natureza modular do GRUB 2 e são carregados necessariamente pelo GRUB 2.

/etc/default/grub

Esse arquivo pode ser editado e configurado. Detalhes da configuração estão mais adiante em Configurando GRUB 2.

/etc/grub.d/ (diretório)

Os arquivos nessa pasta são lidos durante a execução do comando update-grub e

suas instruções são incorporadas ao **/boot/grub/grub.cfg.**

A colocação dos ítens de menu no grub.cfg é determinada pela ordem em que os arquivos são executados nessa pasta. Arquivos com numeral no início são executados primeiro começando pelo menor. 10_linux é executado antes de 20_mentest que é executado antes de 40_custom.

Entradas personalizadas podem ser criadas no arquivo 40_custom ou outro recém criado.

Somente arquivos executáveis geram saída para o grub.cfg durante a execução do update-grub.

Os arquivos default nesta pasta são :

- 00_header: suas configurações normalmente são importadas de informações do /etc/default/grub e os usuários não precisam mudar esse arquivo.

- 05_debian_theme : configura splash image, cor do texto, cor de realce e tema. Na ausência de splash image esse arquivo configura um tema monocromático para o menu inicial.

- 10_hurd : não usada.

- 10_linux : localiza kernels no root device para o sistema operacional em uso. Leva essa informação e estabelece os nomes apresentados no menu para estes kernels.

- 20_memtest86+ : procura por /boot/memtest86+.bin e o inclue como opção no menu do GRUB 2. Não há opção para remover essa entrada do menu.Pode ser inibido removendo o executável desse arquivo sudo chmod -x /etc/grub.d/20_memtest86+ e executando depois update-grub.

- 30_os-prober : procura por Linux e outros sistemas operacionais.Resultados são colocados no grub.cfg baseando-se nos scripts desse arquivo. O arquivo é dividido

em 4 seções representando os tipos de sistemas operacionais manipulados pelos scripts : Linux, Windows, OSX e Hurd. Variáveis nesse arquivo determinam o formato dos nomes exibidos no /boot/grub/grub.cfg e no menu do GRUB 2. Usuários familiarizados com scripts básico podem alterar essas variáveis e mudar o formato das entradas de menu exibidas. Alternativamente o usuário pode inserir uma entrada no /etc/default/grub a qual desativa esse script.

40_custom : Para adicionar entradas de menu personalizado que serão inseridas no grub.cfg depois de update-grub. Mais informações sobre esse arquivo na parte de configuração.

16.2.4 Configuração

Alterações na configuração são normalmente feitas em /etc/defaul/grub e nos arquivos da pasta /etc/grub.d.

O arquivo **/boot/grub/grub.cfg** não deve ser editado pelo usuário; mudanças nesse arquivo são feitas pelos scripts de configuração.

Algumas das alterações mais comuns como OS/kernel default, menu timeout podem ser feitas pelo StartUp-Manager.

O Arquivo **/etc/default/grub** é o principal arquivo de configuração para alterar as configurações padrão.

Após a instalação as seguintes linhas podem ser alteradas pelo usuário :

- GRUB_DEFAULT

Configura a entrada padrão do menu.

Entradas podem ser numéricas ou “saved” (última usada).

- GRUB_DEFAULT=0

Configura a entrada default pela posição. Como no GRUB a primeira entrada é zero, a segunda 1, etc,

- GRUB_DEFAULT=saved

Define a entrada de menu padrão com o que foi selecionado anteriormente (a última seleção).

Se o menu é exibido durante a inicialização, a última entrada selecionada será realçada. Se nenhuma ação for tomada, esta seleção será iniciada no final do tempo limite ou se o menu está oculto.

- GRUB_DEFAULT="xxxx"

Uma entrada de menu exata, incluida entre aspas, também pode ser usada. Neste caso, a localização no menu não importa. Exemplo: GRUB_DEFAULT="Debian Squeeze"

- GRUB_TIMEOUT= Configura o tempo.

- GRUB_HIDDEN_TIMEOUT=0 O menu ficará oculto.

- GRUB_HIDDEN_TIMEOUT=0

O menu não será oculto

- GRUB_HIDDEN_TIMEOUT_QUIET=true

true = nenhuma contagem regressiva é exibida. A tela ficará em branco. false = Um contador será exibido numa tela em branco durante o tempo do GRUB_HIDDEN_TIMEOUT.

- GRUB_GFXMODE=640×480 Descomente essa linha para mudar a resolução. Ela fornece resoluções suportadas pela placa de vídeo do usuário (640×480, 800×600,

1280×1024, etc). aplica-se apenas a resolução do menu de boot.

Dica: Definindo a mesma resolução do sistema operacional o tempo de inicialização diminuirá ligeiramente.

Embora não seja necessário, o usuário também pode especificar a cor BitDepth anexando-o à configuração da resolução. Um exemplo seria 1280x1024x24 ou 640x480x32.

O usuário também pode adicionar várias resoluções. Se o GRUB2 não pode usar a primeira entrada, ela irá tentar o próximo ajuste.

As configurações são separadas por uma vírgula. Exemplo: 1280x1024x16, 800x600x24, 640×480.

Se utilizar uma splash image, certifique-se que a configuração da resolução e do tamanho da imagem são compatíveis.

Se estiver usando uma entrada que produz uma mensagem “não encontrado” ao executar update-grub, tente adicionar ou alterar a cor BitDepth.

Se esta linha está comentada (#) ou a resolução não está disponível o GRUB 2 usa a configuração padrão determinada pelo **/etc/grub.d/00_header.**

- GRUB_DISABLE_LINUX_RECOVERY=true Adicione ou descomente essa linha se não desejar o “Recovery” mode no menu. “Recovery mode” para apenas um kernel, fazer uma entrada especial em /etc/grub/40_custom.

16.2.5 Entradas personalizadas

GRUB2 permite ao usuário criar seleções de menus personalizados que serão automaticamente adicionados ao menu principal quando o comando update-grub é exe-

cutado. Um arquivo 40_custom vazio está disponível em /etc/grub.d/ para uso ou para servir de exemplo para criar menus personalizados. Aqui estão algumas considerações para menus personalizados:

O nome do arquivo determina a ordem das seleções no menu. Nomes numérico são executados primeiro, seguido por nomes alfabéticos.

Entrada 10_linux aparecerá antes de 30_os prober, que será colocado antes de entradas 40_custom, que irá preceder my.custom.entries.

O usuário que desejar que as suas entradas de menu personalizado apareçam em primeiro lugar no menu principal pode nomear a pasta para 06_xxx que suas listas estarão em primeiro lugar no menu.

O arquivo deve ser feito executável :

```
1 # chmod +x /etc/grub.d/filename
```

O conteúdo desse arquivo é adiconado ao **grub.cfg** sem alterações quando o comando **update-grub** é executado.

Ao contrário do grub.cfg, arquivos personalizados podem ser editados a qualquer momento, não são só leitura, e podem ser propriedade do usuário, se ele desejar. Os arquivos personalizados são estáticos. O conteúdo não será alterado, quando novos kernels são adicionados ou removidos.

Se o arquivo personalizado coloca as entradas no topo do **grub.cfg**, o usuário poderá confirmar no **/etc/default/grub (DEFAULT= x)** após executar update-grub. Certifique-se se ainda aponta para a entrada de menu desejada. Para contar, a primeira entrada do menu no arquivo /etc/default/grub é 0.

16.2.6 Criando menus personalizados

O usuário pode editar o arquivo default /etc/grub.d/40_custom ou criar um novo. A maneira mais fácil de criar o conteúdo de um menu personalizado é copiar uma entrada do /boot/grub/grub.cfg. Depois de copiado, o conteúdo do 40_custom pode ser adaptado a vontade do usuário.

De acordo com o arquivo personalizado padrão da amostra (etc/grub.d/40_custom/) as duas primeiras linhas de um arquivo personalizado em /etc/grub.d devem ser:

```
1 #!/bin/sh
2 exec tail -n +3 $0
```

- podem ser copiadas entradas de menu existente no arquivo boot/grub/grub.cfg ou de um arquivo do GRUB legacy. Se for copiado do arquivo menu.lst do GRUB legacy:

- uma cópia de backup deste arquivo pode estar na pasta /boot/grub se o usuário atualizou do GRUB para GRUB 2.

- as seguintes entradas do menu do GRUB legacy devem ser mudadas :

- title é mudado para menuentry. A linha deve terminar com ({)

- root muda para set root=

- kernel passa a ser linux

- qualquer designação de partições (sda4, sdb6) deve ser mudada porque no GRUB legacy a primeira partição é 0 e no GRUB2 é 1 (o primeiro disco continua sendo 0 igual ao GRUB legacy).

16.2.7 Regras para construção de menuentry

- a primeira linha inicia com menuentry e termina com ({)
- o que está entre aspas é o que vai aparecer no menu do GRUB 2. Edite como desejar
- a última linha do menuentry deve ser (})
- não deixe espaços em branco no fim das linhas
- a linha set root= deve apontar para a localização de inicialização do GRUB2 (sdXY)
- referência root da linha Linux deverá apontar para a partição do sistema.

Dica : Se GRUB 2 não encontra o kernel referenciado, tente substituir a UUID pelo nome do dispositivo (exemplo: / dev/sda6).

Exemplo de menuentry :

```
1 ### BEGIN /etc/grub.d/10_linux ###
2 menuentry "Debian Squeeze, Linux 2.6.31-15-generic" {
3 recordfail=1
4 if [ -n ${have_grubenv} ]; then save_env recordfail; fi
5 set quiet=1
6 insmod ext2
7 set root=(hd0,0)
8 search -no-floppy -fs-uuid -set 22290d2e-82c9-48d5-92c1-ce138634eedc
9 linux /boot/vmlinuz-2.6.31-15-generic root=UUID=22290d2e-82c9-48d5
     -92c1-ce138634eedc ro vga=789 quiet splash
```

```
initrd /boot/initrd.img-2.6.31-15-generic
}
#Essa é uma entrada de exemplo para uma partição com Windows:
menuentry "Windows Vista" {
set root=(hd0,2)
chainloader +1
}
```

16.2.8 Configurando fonte e cores

Estas linhas estão contidas em /**etc/grub.d/05_debian_theme.**

A cor é definida pela linha abaixo. A primeira cor é a cor do texto, a segunda é a cor de fundo.

```
set color\_normal=black/black
```

O código abaixo define a cor de entradas selecionadas. A primeira cor é a cor do texto em destaque, a segunda é a cor de fundo da linha selecionada. Se black é a segunda entrada, a linha de destaque será transparente e apenas o texto mudará de cor.

```
set color_highlight=magenta/black
```

O black é a cor de transparência GRUB2 padrão . Altere apenas a cor do primeiro (ou seja, xxxxx /black), se utilizar uma imagem de splash.

Se o segundo valor nesta linha é alterada para uma cor diferente de black a imagem splash será escondida atrás de um fundo de cor sólida.

16.2.9 Segurança no grub2

Para melhorarmos nossa segurança local, uma boa seria colocar senha no grub2, então vamos fazer melhor, iremos colocar uma senha criptografada nele, para isso temos que gerar uma senha criptografada:

```
1 # (echo 123456 ;echo 123456) | grub-mkpasswd-pbkdf2
2 Enter password:
3 Reenter password:
4 Your PBKDF2 is grub.pbkdf2.sha512.10000.1CCB58AE691A20A26872A50FF5D0
5 ED4D1C872F8B3366548C3AA23F1B735B5CB6498B672835C7A1FD3D10E3E8E8677776
6 D71658264789132F72B08E66A4224FCD.C6B51DB03F8665FD6B080EDF502DBAD2A20
7 F3F1992FC878C3CDAE11E4AF82C6EA74A19469A7FB4C7F96275FA7DF3834EB7070EE
8 D9FFAEBC9C84ADB9D272052A7
```

Onde : (echo 123456 ;echo 123456) 123456 é a senha, caso não queira passar a senha no comando execute diretamente e preencha o prompt com a senha quando for solicitado:

```
1 # grub-mkpasswd-pbkdf2
```

Após digitarmos nossa senha ela é criptografada em sha512, aí basta copiá-la e editar o arquivo /etc/grub.d/00_header.

Adicione no final do arquivo a seguinte linha:

```
1 # vim /etc/grub.d/00_header
2 cat << EOF
3 set superusers="user"
4 password_pbkdf2 user grub.pbkdf2.sha512.10000.393F191284EF
5 E4575DCFFE4D939635CFDDF63E254B86F0DB409C0EE6723691D24C0BB7
6 3E0599CE6E39DD22EE5713816D155D0F89EABCDBF0057DEB6DDE953401
```

```
7 .8037149B1F04504F84E019100C56D86816D3859ED7DD557CBEEBD2F95
8 B5177CCC5C5947559092C4A5320E70B8AA07C81EEAE37711763589D525
9 F77C54E10CF85E0F
```

Atualize o grub2:

```
1 #  update-grub
```

A importância de termos senha no Grub é que se o mesmo estiver livre de senha, qualquer pessoa na hora da inicialização pode editá-lo e inicializá-lo para ganhar poderes de root sem saber

16.3 Colocar Imagem no Grub2

Para colocar uma imagem no grub2 é bem simples basta ter uma imagem valida, JPEG, PNG, TGA e JPG, e passar o seu caminho dentro de um arquivo e ja esta pronto. Vamos a prática, dentro do diretório home tem uma imagem jpeg, dexter.jpeg, vamos colocar ela no grub. Copie a imagem para o /boot/grub:

```
1 cp /root/dexter.jpeg /boot/grub
```

Entre no arquivo 05_debian_theme na linha 146 e mude a imagem padrao do grub para a nova:

```
1 # vim +146 /etc/grub.d/05_debian_theme
2 Na linha onde
3 ANTES
4
```

```
5  145 set_background_image "${WALLPAPER}" "${COLOR_NORMAL}" "${
      COLOR_HIGHLIGHT}" |    |
6  146 set_background_image "/usr/share/images/desktop-base/desktop-
      grub.png" ||
7  147 set_default_theme
8
9  DEPOIS
10
11 145 set_background_image "${WALLPAPER}" "${COLOR_NORMAL}" "${
      COLOR_HIGHLIGHT}" |    |
12 146 set_background_image "/boot/grub/dexter.jpeg" ||
13 147 set_default_theme
```

Pronto, agora basta atualizar o grub, ja vai estar valendo:

```
1 # update-grub2
```

16.4 Atualizando novas entradas no menu

Execute o comando para inserir o novo "kernel" no arquivo de configuração do GRUB ou edite-o na mão:

```
1 # update-grub2
```

OU

```
1 # vi /boot/grub/grub.cfg
```

```
1  menuentry 'Debian GNU/Linux, with Linux 2.6.32-5-686' --class
   debian --class
gnu-linux --class gnu --class os {
2          insmod part_msdos
3          insmod ext2
4          set root='(hd0,msdos2)'
5          search --no-floppy --fs-uuid --set 8970340d-ac56-461b-815a
   -2388f9bdadd3
6          linux   /vmlinuz-2.6.32-5-686 root=UUID=661d7f7c-3bfc-4b9e
   -b6d3-9c56cda87d3a ro  quiet
7          initrd  /initrd.img-2.6.32-5-686
```

Após feitas essas alterações, reinicie o computador, e veja se o nosso Novo kernel "boota".

Para adicionar automaticamente um kernel ao Bootloader o nome do kernel deve começar com vmlinuz e as imagens dos módulos com initrd ou initramfs.

```
# update-grub
ou
# update-grub2
```

```
1 # grubby --title=CentOS --add-kernel=/boot/vmlinuz-3.2.1-20111225c1
    --initrd=/boot/initrd.img-20111225c1 --args="root=/dev/VolGroup/
    lv_root"
```